

CTI980101

CONTRACT No. USZA22-97-P-0051

HELMSMAN'S RECORDING ACCELEROMETER

Prepared by:

Donald F. DeCleene, Principal Investigator
Dennis A. Silage, Principal Electrical Engineer
Mitchell B. Oslon, Research Engineer

Conrad Technologies, Inc.
Station Square One, Suite 102
Paoli, PA 19301

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

January 1998

PHASE I FINAL REPORT

19980520 123

Prepared for:

United States Special Operations Command (USSOCOM)
ATTN: CDR. John Stenard, SOAC-M&R-NSW RIB
7701 Tampa Point Boulevard
MACDILL AFB, FL 33621-5323

DTIC QUALITY INSPECTED 3

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate to any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE 4 Jan 98	3. REPORT TYPE AND DATES COVERED Final Report 4 Jul 97 - 4 Jan 98
4. TITLE AND SUBTITLE Helmsman's Recording Accelerometer			5. FUNDING NUMBERS USZA22-97-P-0051
6. AUTHOR(S) Donald F. DeCleene Dennis A. Silage Mitchell B. Oslon			8. PERFORMING ORGANIZATION REPORT NUMBER 980101
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Conrad Technologies, Inc. Station Square One, Suite 102 Paoli, PA 19301			
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) United States Special Operations Command (USSOCOM) Attn: Cdr. John Stenard, SOAC-M&R-NSW RIB 7701 Tampa Point Boulevard Macdill AFB, FL 33621-5323			10. SPONSORING /MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES			
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited.			12b. DISTRIBUTION CODE
13. ABSTRACT (Maximum 200 words) <i>Report developed under SBIR Contract</i> Special Operation Forces (SOF) must meet mission requirements in all types of sea conditions. High performance craft, forced to operate in rough sea, create rapid onset of high levels of fatigue for craft occupants, resulting in reduced mission performance and creating a higher potential for injury of the craft occupants. The objective of this program is the development of a Helmsman's Recording Accelerometer (HRA) for instrumentation on high performance craft. The HRA is needed to provide the helmsman with real time quantitative acceleration data and to record craft operation data that will allow for post-mission analysis. A Phase II program is required for implementation of the HRA. An HRA test unit, fabricated to meet Phase I objectives, requires operational testing to ensure that the HRA provides data that is beneficial and optimally formatted for the helmsman in controlling the craft. Field tests are essential to establish post-mission database requirements for the HRA and final HRA design requirements. The database is needed to evaluate passenger safety, to develop improved mission operations, to obtain craft performance rating, to identify craft and personnel performance limits, and for the development of effective maintenance. The HRA database will serve to enhance future mission planning.			
14. SUBJECT TERMS SBIR Report Data Acquisition Navigation Seaworthiness Recording Accelerometry			15. NUMBER OF PAGES 84
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT

ACKNOWLEDGEMENT

The guidance given by Cdr. John Stenard of USSOCOM during the course of the effort leading to this report is gratefully acknowledged.

EXECUTIVE SUMMARY

Special Operation Forces (SOF) are required to respond rapidly to mission requirements with high performance craft operating in all types of sea conditions. Mission requirements and sea conditions may be such that severe operational demands are placed on SOF personnel. Acceleration loading on the craft and craft personnel in rough water wave action create rapid onset of high levels of fatigue, resulting in reduced mission performance and creating a higher potential for injury of the craft occupants.

It is difficult for the helmsman to judge the wave impact upon the craft, and consequently, to take corrective action for the optimal implementation of mission requirements and for the appropriate protection of the craft occupants. Real time display of vertical acceleration would enable the helmsman to obtain a more immediate and more reliable judgment of the behavior of the craft. This would allow the helmsman to select headings and speeds that would reduce the acceleration loading and injury potential of the wave action. Moreover, a database to archive craft acceleration histories and craft operational data for each mission would allow for improved post-mission analysis and serve to enhance future mission planning, command, performance and safety. Such a database is needed to evaluate helmsman and passenger safety, to develop improved mission operations, to obtain craft performance rating, to identify craft and personnel performance limits, and for the development of improved, and more cost effective, maintenance and inspection procedures.

The objective of this program is the development of a Helmsman's Recording Accelerometer (HRA) for console instrumentation on MSW RIB and MK V SOC. The HRA is to display craft acceleration data at the console. This is to provide the helmsman with data that assists the helmsman in choosing speeds and headings for the craft in order that the helmsman is able to reduce the perceived roughness and fatigue placed upon the craft and craft personnel. The HRA is to record the mission data to allow for post-mission analysis of the craft mission and operation.

The design and fabrication of an initial HRA test unit has been completed and is available as a result of the Phase I effort of the HRA program. It is recommended that this HRA unit be tested and evaluated in actual field operations in order to further identify, and quantify, features for the HRA that serve to optimize the HRA design for use by the helmsman with the high-performance craft. Testing and evaluation is recommended to ensure that the final design of the HRA meets all system and operational requirements, including such items as the optimal format for the display of information to the helmsman. In addition, it includes the development of procedures for optimizing the operator's selection of operating modes and for the development of a practical command interface. It also includes development of the type, extent and presentation of information required for improved command and control of the craft and for post-mission analysis. Furthermore, it is recommended that craft operating information useful for improving the performance and maintenance of the craft, such as engine boost pressure and engine RPM, be evaluated and incorporated into the HRA data recording.

TABLE OF CONTENTS

EXECUTIVE SUMMARY	i
1.0 INTRODUCTION.....	1
2.0 CONCLUSIONS AND RECOMMENDATIONS	2
2.1 Conclusions.....	2
2.2 Recommendations	3
3.0 PHASE I DEVELOPMENT.....	5
3.1 Purpose and Scope of the Research Effort	5
3.2 Results of Individual Tasks	5
3.2.1 Task 1: Review Functional Requirements	6
3.2.2 Task 2: Establish HRA System Requirements.....	6
3.2.3 Task 3: Finalize Fabrication and Assembly Details of the HRA	6
3.2.4 Task 4: HRA Fabrication and Assembly	7
3.2.5 Task 5: HRA Design Verification and Calibration Tests	7
3.2.6 Task 6: Application and Installation Support for the HRA	7
3.2.7 Task 7: Preparation of Deliverables.....	7
4.0 PARAMETERS AND SPECIFICATIONS	8
4.1 Helmsman Recording Accelerometer (HRA).....	8
4.2 Data Acquisition Component	9
4.3 Console Display Component	10
4.4 Operator's Input Panel Component	11
4.5 Data Transfer Adapter	11
4.6 Acceleration Sensors	11
4.7 Inclinator and Electric Compass Component	12
4.8 Cables and Connectors	12
4.9 Transportable Data Storage Unit	12
4.10 Post-mission Analysis Computer	12
4.11 Auxiliary Boat Instrumentation Data	12
5.0 HRA TEST UNIT DESCRIPTION.....	13
5.1 Data Acquisition Component	13
5.2 Console Display and Operator's Input Component.....	14
5.3 Data Transfer and Display Connector Assembly	14
5.4 Triaxial Accelerometer Component	15
5.5 Inclinator and Electronic Compass Component	15
5.6 Cables and Connectors	15
5.7 Auxiliary Boat Instrumentation Data	15
5.8 Portable Computer	16
6.0 INSTALLATION INSTRUCTIONS	16
7.0 OPERATING INSTRUCTIONS.....	20
8.0 REFERENCES	22

APPENDIX A: FIGURES

APPENDIX B: DATA FORMAT AND TRANSFER

APPENDIX C: SOURCE CODE

1.0 INTRODUCTION

Special Operation Forces (SOF) are required to respond rapidly to mission requirements in all types of sea conditions. Mission requirements and sea conditions may be such that severe operational demands are placed both on the operation of the craft, the safety of the craft occupants and the fatigue levels of the craft occupants. Accelerations of the craft and acceleration loading on the craft personnel in rough water wave action is known to induce fatigue and injury on the craft occupants and consequently reducing their mission performance.

Sea conditions may be such that it is difficult for the helmsman to judge the wave impact upon the craft, and consequently, to take corrective action for the optimal implementation of mission requirements and for the appropriate protection of the craft occupants. Real time display of vertical acceleration would enable the helmsman to obtain a more immediate and more reliable judgment of the behavior of the craft. This would allow the helmsman to select headings and speeds that would reduce the acceleration loading and injury potential of the wave action. Moreover, a database to archive craft acceleration histories and craft operational data for each mission would allow for improved post-mission analysis and serve to enhance future mission planning, command, performance and safety. Such a database is needed to evaluate helmsman and passenger safety, to develop improved mission operations, to obtain craft performance rating, to identify craft and personnel performance limits, and for the development of improved, and more cost effective, maintenance and inspection procedures.

The HRA is needed as a component of the USSOCOM system for the MSW RIB and MK V SOC. It is needed to optimize the mission performance of the helmsman and the MSW RIB and MK V SOC. The development and fabrication of alternative HRA instrumentation to measure acceleration at various locations on the MSW RIB and/or the MK V SOC is well within the capability of available sensor components and data collection techniques. The design and fabrication of the initial HRA test unit is a significant step toward the development of a valuable instrument for enhancing the mission performance of the helmsman and the MSW RIB and MK V SOC.

The design and fabrication of an initial HRA unit, intended as a prototype test unit, is completed and available as a result of SBIR Phase I program, reference contract USZA22-97-P-0051. This HRA unit needs to be tested and evaluated in actual field operations in order to identify, and quantify, features for the HRA that serve to optimize the HRA design for use by the helmsman with the high-performance craft. This is necessary to ensure that the development of the HRA is properly directed to meet all system requirements, including such items as the human factors related to the training of the helmsman and the type, location and extent of information required on the behavior of the craft.

Craft instrumentation, such as a Helmsman's Recording Accelerometer (HRA) which is designed to measure and display the accelerations encountered by a craft operating at high speed during rough sea conditions, would provide the helmsman with immediate and valuable information on the operation of the craft in the rough water. When properly tested and implemented the HRA system will assist the helmsman in choosing speeds and headings to reduce perceived roughness

during SOF missions and to reduce fatigue and increase safety of craft personnel. Utilizing the HRA for recording data will provide post-mission analysis of mission and craft. Post-mission evaluation of the loading on the craft is important to archive craft performance for various sea conditions, to evaluate safety parameters, and to improve craft performance data for future mission planning.

The selection of an optimal design for the HRA is influenced by the system factors arising from functional, physical and environmental considerations, and the technical challenge for the practical development of the HRA is the assessment of these system requirements and the design alternatives. Specification of the type of data required, including accuracy and/or resolution, has a significant bearing on the optimal design arrangement. The optimal HRA design is defined not by the development of specialized electrical components but rather by its ability to satisfy the functional, operational, physical, environmental, cost and installation factors for the MSW RIB and MK V SOC.

2.0 CONCLUSIONS AND RECOMMENDATIONS

The results of the Phase I program effort leading to the development and fabrication of a Helmsman's Recording Accelerometer (HRA) for console instrumentation on the MSW RIB and MK V SOC are described in this report.

The HRA is needed to reduce the potential for injury to craft personnel and to optimize the mission performance of the helmsman and the craft. It provides a means of significantly improving the command, control and post-mission analysis of the operation of the MSW RIB and MK V SOC.

2.1 Conclusions

Design parameters and specifications for the HRA were established under this Phase I program. The design parameters and specifications were established in accordance with the program objective of providing the helmsman with real time quantitative data and providing a record of craft accelerations. The operational environment, the application requirements and available literature on rough sea operation, coupled with the objective of improving rough water operation and providing for post-mission analysis of craft operation, served as the basis for establishing the specifications.

The design of a Helmsman Recording Accelerometer (HRA) was completed and an initial HRA unit was fabricated in accordance with the design parameters and specifications established under this Phase I program. The components and details of the HRA were selected from an extensive review of alternative electronic components and hardware options, with final selection based on the ability of the components to meet, or exceed, the established specifications. The initial HRA design was established with programmable features and, as such, allows modification of the HRA data collection and display procedures for adaptation to future, and currently undefined, requirements for optimal helmsman's utilization of the HRA.

The initial HRA unit fabricated under this program incorporates, and establishes, the technical conclusions derived in this program. The design and fabrication is formulated in accordance with the parameters and specifications defined by the application, operation and literature review accomplished under this program. The HRA unit provides a practical approach to improving mission performance of the MSW RIB and MK V SOC. The HRA unit fabricated under this program is available for demonstration and mission testing.

The parameters and specifications established in this Phase I program are presented in Section 4.0 of this report. The final design arrangement is calibrated and tested through in-house testing of the HRA unit. Operating procedures for the HRA, established for initial field test parameters, are presented in Section 6.0 of this report. The source code established for operation of the HRA is presented in Appendix C of this report.

2.2 Recommendations

The most effective HRA design is defined not by the development of an initial design which meets specifications but rather by the ability of the HRA unit to satisfy the functional, operational, physical, environmental, cost and installation factors for the MSW RIB and MK V SOC. As such, operational testing and adaptation of the initial HRA design is necessary to determine the usefulness of the HRA and to effectively develop and employ the HRA. The initial HRA unit, furnished under the Phase I program, must be properly installed and adapted to the helmsman's environment and needs. It is recommended that the HRA test unit be installed and that a Phase II program be undertaken to fully evaluate and optimize the performance and utilization of the HRA under actual field operations. With this approach, the HRA can be optimized and serve as a valuable instrument for enhancing the mission performance of the helmsman and the MSW RIB and MK V SOC.

Operational testing of the initial HRA unit is needed to optimize the features of the HRA system. Operational testing is needed to identify additional HRA features that might serve to enhance the use of the HRA. The testing is required for the evaluation of the features that are already incorporated into the HRA system and the identification of the features, functions, scale, information or similar parameters which can serve to enhance the helmsman's control of the high performance craft.

The computer processing unit and the programmable features of the display incorporated into the initial HRA test unit provide for easy modification of the display and operating procedures of the current design. The programmable features of initial HRA test unit allow testing and evaluation of alternative display formats in response to the results of the recommended field testing and in response to recommendations resulting from the helmsman's interpretation of the HRA data display and data collection. To properly utilize this programmable feature of the initial unit, it is recommended that operational testing be accomplished to allow for the evaluation of alternative helmsman's display procedures and for alternative computational procedures for the collection and utilization of post-mission data. The type, extent, formatting, display and analysis of HRA data required for improved command and control of the craft and for post-mission analysis are

factors to be established in the recommended Phase II program. The ability of the HRA test unit to be reprogrammed and provide evaluation of alternative system procedures provides an opportunity to optimize the HRA system through appropriate operational testing.

A Phase II program comprising the following requirements is recommended:

1. Operational testing and application development of the initial HRA design is required. This includes the installation of the HRA and the collection of data on craft operations and system performance. Evaluation of the existing operational parameters of the HRA, evaluation of how the HRA is utilized by the helmsman, and identification of the opportunities for utilization of the HRA data by the SOF command is to be accomplished. The evaluation needs to be based on the data collected on the craft performance and on the user's response to the presentation of the data. The evaluation is to be accomplished with data obtained from operational testing of the HRA unit under an established set of operating conditions and mission requirements. This testing and evaluation requires the installation of the initial HRA test unit onto one or more selected craft. It also requires the interfacing of auxiliary boat instrumentation with the HRA test unit and the collection of operational data from the auxiliary boat instrumentation. Additional HRA units are to be provided if required for complete operational testing.
2. The HRA test unit is required to be installed and programmed to archive the entire performance history of a representative set of missions under varied operational conditions. The data to be archived will include data from the HRA instrumentation and auxiliary craft instrumentation. Documentation of sea conditions, wave action and wave onset, preferably through photographs of sea condition, is required to correlate HRA data with sea action. The data collected on sea condition, mission performance and craft operation is to be documented for SOF review. Command and control procedures for improving the operation and mission planning of the MSW RIB and MK V SOC is required based on analysis and evaluation of the collected data.
3. Based on the Helmsman's response, interpretation, and utilization of the displayed data, and/or based on the need for the display of additional or alternative data as determined by the field tests, the Phase II program effort is required to include reprogramming of the process computer and helmsman display. This is required to respond to revised data collection requests by SOF and/or craft helmsman. Programming of the programmable data display is required to evaluate alternative data display formats and to evaluate the helmsman's ability to utilize alternative data formats. Revised data procedures are to be tested and evaluated in subsequent field tests.
4. Mission planning, operational command and operational control procedures are to be identified from the data collected by the testing program. The complete inventory of data collected is to be made available for post-mission analysis of the mission and craft operation. The complete inventory of the data collected will be used to select the data which is useful in post mission analysis and in identifying mission performance. The collected data will be evaluated to identify appropriate procedures for post mission analysis. Procedures to utilize

the data for improving mission planning, operational control and command control will be identified. Appropriate computer processing procedures for the selection and post-mission analysis of collected data will be established.

5. Post mission analysis of the data collected will be accomplished to identify mission performance, identify parameters for future mission planning and improve helmsman control procedures. The performance capabilities of the craft, the overall ability of the HRA to provide the helmsman with needed, or desired, operational information and the ability of the helmsman's to utilize HRA data under varied sea conditions will be evaluated through post mission analysis of the operational test data. Post mission analysis will also be utilized to identify sea conditions, the craft's ability to maintain speed and time to destination, the impact of sea roughness on craft personnel, and the overall ability of the HRA to provide the helmsman with desired operational information.
6. The development of a finalized HRA design will be accomplished to satisfy the requirements established by the testing and evaluation and to minimize the fabrication cost of the prototype and production units. Delivery of a pre-production prototype HRA unit in accordance with the finalized design is required.

3.0 PHASE I DEVELOPMENT

3.1 Purpose and Scope of the Research Effort

The technical objective of this program is the development of a Helmsman's Recording Accelerometer (HRA) for console instrumentation on MSW RIB and MK V SOC. The HRA is to provide the helmsman with real time quantitative acceleration data and to record craft acceleration data for assistance in post-mission analysis and improvement of rough water operation. The HRA is to be capable of displaying and recording selective operational information of the boat, such as boat attitude, that is determined within the Phase I program as being useful and within practical cost limits for the HRA. The program effort under Phase I included the assembly of a prototype HRA developed by CTI and further included technical support for installation of the HRA.

The acceleration data displayed by the HRA at the console is to provide the helmsman with data that assists the helmsman in choosing speeds and headings for the craft and allows the helmsman to reduce the perceived roughness and fatigue placed upon the passengers and hull structure. The data recorded by the HRA is provided to allow post-mission analysis of the craft mission and operation.

3.2 Results of Individual Tasks

The technical objectives of this program were undertaken in the form of a multi-task program as outlined in the proposal. The hardware references presented in the Phase I proposal and the hardware development procedures presented in the task descriptions of the proposal were used to

direct the development of the HRA. Final hardware selection and details were developed in accordance with HRA system parameters established under this Phase I program. The Phase I development effort of this program is described further under the individual tasks presented below.

3.2.1 Task 1: Review Functional Requirements

The functional requirements for the HRA were reviewed by CTI. System interface requirements, operating limits, and overall user requirements were identified and presented for review by SOCOM. The identification of HRA design requirements was coordinated by CTI with SOCOM and the boat manufacturer.

The effort under this task included a review of the anticipated operational parameters for the HRA, such as craft operation, mission requirements and data collection requirements. It included a review of the system parameters affecting the selection of an arrangement for displaying acceleration information to the helmsman. It included a review of the operational and environmental factors that influence structural fatigue and the corresponding need for structural fatigue monitoring. Numerous design and hardware alternatives for data collection, data storage, data transfer and post-mission analysis were reviewed to determine hardware requirements needed for an optimal HRA system. An evaluation of the requirements for collection, recording, transferal, analysis and use of the acceleration data obtained with the HRA was accomplished. A review of literature pertaining to human fatigue in a low frequency acceleration or vibration environment such as that encountered with small craft in rough sea was also accomplished.

Application information provided by SOCOM, along with any additional information available under this program, was utilized in determining the final specifications for the design of the HRA. Information was provided to the USSOCOM RIB Acquisition Office in the progress reports. The findings of this task were discussed with SOCOM and utilized in the development and selection of the final design arrangement for the HRA.

3.2.2 Task 2: Establish HRA System Requirements

HRA system requirement and specifications that need to be satisfied by the embodiment of the HRA design were identified and established as requirements for the design. The system requirements were established in consultation with SOCOM. Cost-benefit trade-offs of alternative arrangements were evaluated by CTI to ensure that the programmatic goals of USSOCOM would be met.

3.2.3 Task 3: Finalize Fabrication and Assembly Details of the HRA

The fabrication and assembly details for fabrication of the HRA were completed under this task. The development of the HRA was accomplished in accordance with the results and information established by Tasks 1, 2 and 3. Specifications were identified for components of the design. Electronic circuit board layouts, electronic schematics, assembly sketches, detail drawings, and related design specification requirements were accomplished as needed to complete the

fabrication of the prototype HRA. Mechanical details of the customized display case and assembly hardware were drawn in AutoCad (Autodesk, Sausalito, CA).

3.2.4 Task 4: HRA Fabrication and Assembly

An HRA prototype unit was fabricated under this Task. A description of the fabricated unit is presented below. The HRA was bench tested and the Data Recovery and Display subsystems were tested and verified.

3.2.5 Task 5: HRA Design Verification and Calibration Tests

The HRA was subjected to repeated mechanical shock administered by impulse and monitored by small, lightweight accelerometers temporarily mounted on the components. The acceleration in three axes had a peak magnitude of at least 10g. Low frequency vibration of the HRA was administered by electromechanical translation. The electronic calibration of the HRA was performed by a digital frequency counter for the real-time clock and digital voltmeter for the analog-to-digital converter.

3.2.6 Task 6: Application and Installation Support for the HRA

The application and installation of the HRA in the field is guided by the documentation of the HRA provided in this report. This includes application of the HRA for data acquisition, retrieval, processing, and display. The utilization and value of the HRA system can be enhanced further by field-testing and by selecting and organizing the collection of the data to specific requirements of mission analysis. Data recorded by the HRA can be used to support craft design decisions, maintenance activities and post-mission analysis. A continuation of the development of the HRA, which would include development of the recommendations provided by this report, was originally scheduled for development and implementation under a proposed Phase II effort for this program. Section 2.2 of this report provides recommendations for a Phase II program to enhance the utilization and application of the HRA.

Proper mounting of the HRA on the boat was identified in this task. Technical support for the installation of the HRA was provided by visiting the boat manufacturer, inspecting the craft for installation requirements and taking measurements of the craft to fabricate custom lengths of cable for installation of sensors and HRA components. The installation instructions provided in this report present the results of this task.

3.2.7 Task 7: Preparation of Deliverables

Interim documentation, in the form of monthly letter reports, was submitted during the course of the project to summarize the status and progress of the work during the reporting period. This report serves as the final report and includes the development details and overall results of the Phase I program.

4.0 PARAMETERS AND SPECIFICATIONS

The parameters and specifications for the HRA were established by CTI from a review of the physical characteristics of the MSW RIB and MK V SOC, the operating characteristics of the craft, the operational environment, the helmsman's need for acceleration data and the requirement for data for post-mission analysis. Requirements presented in the SBIR announcement, craft data obtained from the boat manufacturer, requirements presented by SOCOM and information obtained through literature review serve as the basis for establishing the parameters and specifications under Tasks 1 and 2 of this project.

Strain instrumentation and strain data collection for the statistical analysis of structural fatigue was evaluated by CTI in cooperation with SOCOM and the boat manufacturer. Inspection of the craft at the manufacturer indicated that the craft exhibited very high strength and durability. Based on the inspection, and with the craft being identified as exhibiting both high strength and durability, the recording of stress-strain loading for structural fatigue analysis was determined to be unnecessary for the MSW RIB and MK V SOC. The mission data recorded by the HRA concerning craft operation and vertical acceleration of the craft was selected as sufficient for long term analysis of the hull accelerations, hull structure loading and hull structure fatigue.

The parameters and specifications that have been established for the development of the HRA and for the components of the HRA are presented below.

4.1 Helmsman Recording Accelerometer (HRA)

The HRA system is to provide the helmsman with real time quantitative acceleration data of the MSW RIB and MK V SOC during mission operation. The HRA is to have a display that can be mounted on the helmsman's console and provide for visual observation of displayed information by the helmsman. The HRA is to record and store acceleration data during mission operation in a non-volatile internal memory and, upon command, provide for the transferal of the stored data to a transportable data storage unit. The transportable data storage unit is to provide a means of storing and allowing for the transferal of data for post-mission analysis. The HRA is to operate from the available on-board power of +24 VDC. Vertical acceleration of the craft is estimated at 2.5g average and 5g maximum.

Figure 1 shows the specification for the general arrangement of the HRA. A complete HRA system must include the following components:

- Data Acquisition Component
- Console Display Component
- Operators Input Component
- Data Transfer Adapter
- Acceleration Sensors
- Inclinator and Electric Compass Component

- Cables and Connectors

The HRA must be compatible with and allow operation with system components as follows:

- Transportable Data Storage Unit
- Post-mission Analysis Computer
- Auxiliary Boat Instrumentation Data

4.2 Data Acquisition Component

The data acquisition component is required to collect, process, record and make-available data useful for operation of the boat by the helmsman and for post-mission analysis of craft operation and mission. The data acquisition component must be packaged and sealed for continuous operation in a marine environment.

Sixteen analog input channels, two high-speed serial communication data ports, a bidirectional parallel data port, and a four-bit binary data port are required for the data acquisition component. Twelve of the available sixteen analog channels are to be available to interface the accelerometers, electronic compass, and inclinometers to the HRA. One of the serial data ports is to provide access for the data transfer function. The remaining serial data port is available to interface to the National Marine Electronics Association (NMEA) standard GPS receiver (Furuno 1600F) and marine radar (Furuno 841) if deemed a requirement under a Phase II assessment to record such data for post-mission analysis.

The four remaining analog channels and the binary data port are required to allow interfacing of the mechanical trim tab and the engine (Caterpillar 3126) boost pressure and tachometer if recording such data for post-mission analysis is deemed a requirement under a Phase II assessment.

Within the data acquisition component, the data processing subassembly is required to provide analog-to-digital signal conversion, digital filtering, and subsequent data compression and storage for each available sensor. The processing system is required to continuously interrogate the sensors and storage device, providing feedback and providing control of the entire data acquisition, data display and data storage processes. Additionally, the processing system must continuously update the console display with current sensor/system output. Finally, the processing unit is required to enable handshaking between the HRA and the external post-mission analysis computer through the data transfer connection port.

The data acquisition component is required to have external cable connectors for the connection of the remote components of the HRA system. The external cable connectors must be waterproof connections. Connectors are required to allow connection of a 24VDC power, to allow connection of remote sensors, to allow connection of the console display component, to allow connection of the operator's input component, and to allow connection of a data transfer unit or small computer for post-mission transfer of collected data.

4.3 Console Display Component

Visual display of HRA data to the helmsman requires a console display component. The console display component, furnished with the first HRA unit, is to be a commercially available, sunlight-readable, liquid crystal display (LCD). The LCD must be mounted in an enclosure to provide vibration and weather protection. This display component must be fabricated as a stand-alone component. The stand-alone enclosure for the display must allow a non-permanent installation on the console without requiring significant modification of the console to mount the HRA display component. With this stand-alone arrangement, the display assembly is to be tried at various positions on the console before establishing a final, or optimal, installation site.

The console display component is required to be a programmable display and allow for the reprogramming and evaluation of various display formats as a means of identifying an optimal display format under a Phase II operational testing program. The optimal format must be developed and selected in Phase II based on the boat operator's satisfaction with specific display alternatives and based on operational testing of the boat with alternative display formats. The display formats, or modes of operation, for the initial programming of the display is to include a power-on display format and preprogrammed display alternatives which can be selected by the helmsman as an alternative to the power-on display. The console display component furnished under Phase I of this program is to be programmed to provide an initial power-on display format and helmsman selected alternative formats as described below.

Pre-programmed display formats are to be provided by CTI for selection by the helmsman. The programmable display must allow for reprogramming of these operator-selected displays. The programmable display, in combination with the operators input panel described below, is to allow for the selection, testing and reprogramming of various display formats. An optimal format is to be based on the boat operator's satisfaction with specific display alternatives and based on operational testing of the boat with alternative display formats. The optimal format is expected to be developed and/or selected under a proposed Phase II development program.

- Continuous "Power-On" Display: The HRA is required to be programmed to always present the power-on display format whenever the HRA is powered and provided the helmsman has not selected an alternative display. For easy recognition of display information by the helmsman, the power-on display format must provide simple bar graphs of real time vertical acceleration for both the Helmsman's location on the boat and the rear passenger location. The orientation of vertical acceleration is defined as the acceleration of the boat perpendicular to the deck.
- "Azimuth/Heading" Display: The helmsman's selection of this display format is required to initiate a display of the inclinometer and the electric compass data for the operator's reference and for system performance verification of the mission information being recorded by the HRA.

- “Scale” Display: The helmsman’s selection of this display format is required to initiate a change in the real-time acceleration display gain and thereby provide the helmsman with the ability to expand or retract the scale of the displayed acceleration in response to various operating conditions.
- “System” Display: The helmsman’s selection of this display format is required to evoke a second function menu with “Start Recording” and “Start Data Transfer” for data acquisition during mission operation and post-mission data transfer.

4.4 Operator’s Input Panel Component

The operator’s input panel is to consist of four, fully sealed, flush-mounted, shock rated, manual switches. The functions of these switches are to be programmable. The programmable parameter is to allow for the testing of the switch functions as originally provided and to allow system reprogramming if desired. If reprogramming is determined to be needed or desirable by the boat operator in the Phase II testing program, the programmable feature of the input panel will allow for appropriate reprogramming of the switch function. The programming for the switches, and the legend provided for the switches, as initially provided, is to provide for helmsman selection of “Azimuth/Heading”, “Scale”, and “System” display format.

The programming of the switches is to allow operator selection of the various display formats during boat operation. When depressed, the selected display mode will remain active for a pre-selected period of time, after which the display will return to the standard operating mode.

4.5 Data Transfer Adapter

A data transfer adapter must be provided to allow high-speed transfer of processed acceleration and other operational data to a transportable data storage unit. The data transfer adapter is required to provide environmental protection for the electrical transmission of data and to provide easy access for the connection and transfer of data to the transportable data storage unit.

4.6 Acceleration Sensors

The HRA system requires three triaxial linear accelerometers. Each linear accelerometer axis is required to provide a range of 0 to 25g. Each triaxial accelerometer assembly must be separately packaged within a waterproof enclosure. Each accelerometer package must have an external waterproof cable connector to allow cable connection with the data acquisition component. The intended locations for mounting the accelerometers are the center of the boat at the helmsman location, at the right rear passenger location and at the left rear passenger location. The triaxial accelerometers are required to be properly oriented with the deck of the craft in order to accurately sense vertical and sway acceleration of the craft.

4.7 Inclinometer and Electric Compass Component

A two-axis (pitch and roll) inclinometer and electric compass component are required by the HRA system. The inclinometer is required to provide $\pm 50^\circ$ minimum dynamic range of measurement of pitch and roll. The inclinometer and electric compass assembly must be packaged within a waterproof enclosure. The inclinometer and electric compass component must have an external waterproof cable connector to allow cable connection with the data acquisition component. The axes of the inclinometer and compass are to be properly oriented with the deck and centerline of the craft to accurately sense the orientation of the craft.

4.8 Cables and Connectors

Electrical cables and connectors are to be provided for installation of the HRA components furnished by this Phase I program. The cable and connectors must provide ease of installation and environmental protection for the electrical transmission of power and signals.

4.9 Transportable Data Storage Unit

A transportable data storage unit is required to collect the processed data from the HRA data acquisition component mounted on the boat. A small computer or an external embedded microcomputer can be used for this function. The HRA unit must be programmed such that a commercially available portable computer can be used as the transportable data storage unit. This portable computer is not provided as a deliverable component of the HRA.

4.10 Post-mission Analysis Computer

The processed data provided by the HRA is required to be compatible with and allow post mission analysis and reprocessing of data with a commercially available computer. The computer is not provided as a deliverable component of the HRA. Computer programs and systems for post-mission analysis are expected to be developed under a proposed Phase II program.

4.11 Auxiliary Boat Instrumentation Data

Auxiliary boat instrumentation data is that data which is obtained from instrumentation, or sensors, mounted on the boat by the boat manufacturer and providing useful data for post-mission analysis on the operation of the boat. Auxiliary boat instrumentation data identified to date as being useful for post-mission analysis include GPS output, engine boost pressure, and trim position. This data need not be displayed. Auxiliary boat instrumentation is not provided as deliverable components of the HRA. The data acquisition component must provide connectors, processing and recording capability to allow for the recording of a limited amount of data provided by auxiliary boat instrumentation. Technical support is to be provided for coupling the auxiliary boat instrumentation with the HRA system.

5.0 HRA TEST UNIT DESCRIPTION

The Helmsman Recording Accelerometer (HRA), pictured in Figure 2, developed and fabricated under this Phase I program includes the following:

- Data Acquisition Component
- Console Display and Operators Input Component
- Data Transfer and Display Connector Assembly
- Triaxial Accelerometer Components
- Inclinometer and Electronic Compass Component
- Cables and Connectors

The HRA developed and fabricated under this Phase I program is compatible with and allows operation with HRA system components as follows:

- Portable Computer for Data Collection and Post-mission Analysis
- Auxiliary Boat Instrumentation Data

5.1 Data Acquisition Component

The Data Acquisition Component furnished under this Phase I program consists of an embedded microprocessor-based computer to collect, process, record and make-available data useful for operation of the boat by the helmsman and for post-mission analysis. The microprocessor-based computer is assembled within a custom fabricated enclosure that is designed to minimize the affect of shock and vibration on the computer assembly. The enclosure is an environmentally sealed aluminum instrumentation case. The internal computer and data storage memory are wired within the sealed enclosure. Connection to external sensors and power is accomplished through positive locking, environmentally sealed connector jacks located on the face of the enclosure as shown in Figure 3. A view of the data acquisition and processing electronics within the enclosure of the Data Acquisition Component is shown in Figure 4.

The embedded microprocessor-based computer of the processing unit utilizes an AMD 486DX, 100 Mhz clock, microprocessor with 4 MB of system RAM, a high baud rate, RS-232C serial communication port, and a PC/104 expansion bus (WinSystems, SAT-DX4-100). The 12 channel analog sensor suite is interfaced to two 12 bit, 8 channel, differential input, analog-to-digital converter peripherals on the PC/104 expansion bus (WinSystems, PCM-A/D12). Non-volatile memory storage is provided by a 32 MB (unformatted) flash disk memory PC/104 expansion bus peripheral, which requires no battery backup (WinSystems, PCM-FLASH). The flash memory module specifically features an extended temperature range (-40°C to +85°C) which will facilitate data retention under adverse conditions. A custom digital electronic module interfaces the computer to the console display and also provides a crystal controlled, real-time clock for data sampling. The module is connected to the 24 bit parallel bidirectional digital data port of the computer.

A high resolution video graphics adapter (WinSystems, PCM-FPVGA), although not required for the vertical acceleration display and recording functions of the HRA, is resident as a PC/104 expansion bus to facilitate a remote session and data file transfers with an external (laptop) computer system using a standard communication application (Symantec, pcANYWHERE).

The data acquisition component operates on +24 VDC at 0.55 A and utilizes two DC to DC converters to provide +5V and ± 12 VDC. Connectors for power input, serial data communication, the sensors, and display are mounted as shown in Figure 3.

5.2 Console Display and Operator's Input Component

The Console Display and Operator's Input Component furnished under this Phase I program consists of a sunlight-readable liquid crystal display (LCD) and environmentally sealed operator input switches assembled into a single watertight enclosure. The enclosure is a custom fabricated, environmentally sealed aluminum case that minimizes the affect of shock and vibration. Connection of the console display and operator's input component to the data acquisition component is accomplished through a 25 conductor cable which terminates with a positive locking, environmentally sealed connector plug. The LCD, pushbuttons, and associated power converters and connectors are wired within the sealed enclosure. The console display and operator's input component, with cable and connector, is shown in Figure 5.

The console display is an electroluminescent 4.7 by 2.5 inch LCD (AND, 1741MST) programmable display. The display operates on +24 VDC at 0.25 A and utilizes a DC to DC converter to provide +5 VDC and a DC to high voltage AC inverter (Endicott Research, E1118) for the electroluminescent panel. The LCD viewing contrast is digitally adjustable under application software control and the variable, negative DC voltage is provided by an integrated circuit (Maxim, MAX749).

The console display and operator's input component has four environmentally sealed push-buttons (MGR, 1111) mounted on the face of the component for operator's input. Operational modes and operator's selection of display format is in accordance with the parameters and specifications presented in section 4.0 above. The display is programmable to allow development of display data and procedures as determined by the proposed Phase II program.

5.3 Data Transfer and Display Connector Assembly

The Data Transfer and Display Connector Assembly furnished under this Phase I program consists of two positive locking, environmentally sealed connector jacks mounted on a bulkhead plate and a 26 pin and 10 pin cable and connector as shown in Figure 6. The two cables are connected to the Data Acquisition Component. The cable from the Console Display and Operator's Input Component is connected to the 26 pin connector jack. The 10 pin connector jacks is the Data Transfer port to retrieve stored data from the HRA.

5.4 Triaxial Accelerometer Component

Three identical triaxial accelerometer components are furnished under this Phase I program. Each of the components consists of a micromachined triaxial accelerometer (Crossbow, CXL25M3) mounted in an environmentally sealed, aluminum case (Hammond, 1590AF). The triaxial accelerometers operate on +24 VDC at 0.015 A and utilize a DC to DC converter to provide +5 VDC. The accelerometers have a full scale span of ± 25 G in each axis. Ten (10) pin, positive locking, environmentally sealed connector jacks are located on the face of the enclosure to provide for cable connection to the data acquisition component. The triaxial accelerometer components furnished under this Phase I program are shown in Figure 7. There are three such triaxial accelerometers, one is to be mounted below deck at the helmsman position and two mounted below deck, aft starboard and aft port.

5.5 Inclinator and Electronic Compass Component

The Inclinator and Electronic Compass Component furnished under this Phase I program consists of a two axis tilt sensor and a triaxial magneto-inductive magnetometer with an integral, compensating two axis tilt sensor. A triaxial magneto-inductive magnetometer with an integral two axis tilt sensor provides an analog voltage proportional to the compass direction (Precision Navigation, TCM2). Another wide dynamic range ($\pm 60^\circ$), two axis tilt sensor provides an analog voltages proportional to pitch and roll (Fredericks, 0717). The sensor suite operates on +24 VDC at 0.060 A and uses two DC to DC converters to provide +5 V and +12 VDC. The sensor suite is housed in a high impact, non-metallic (ABS plastic) case (Hammond, 1594CW). A (10) ten pin, positive locking, environmentally sealed connector jack is located on the face of the enclosure to provide for cable connection to the data acquisition component. The inclinometer and electronic compass component furnished under this Phase I program is shown in Figure 8.

5.6 Cables and Connectors

The Cables and Connectors furnished under this Phase I program consist of positive locking, environmentally sealed connector plug and jacks and shielded, instrumentation grade (number 22 wire) cable.

5.7 Auxiliary Boat Instrumentation Data

Auxiliary boat instrumentation data is that data which is obtained from instrumentation, or sensors, mounted on the boat by the boat manufacturer and providing useful data for post-mission analysis on the operation of the boat. The Data Acquisition Component, fabricated under this Phase I program, has connectors and data processing capacity for collection of a limited amount of auxiliary boat instrumentation data as follows:

- Analog channel 13 is available for the collection of mechanical trim tab position data.

- Analog channel 14 is available for collection of engine (Caterpillar 3126) boost pressure data.
- Analog channel 15 is available for collection of engine tachometer output.
- Serial port COM2. This serial data port is available to interface to the National Marine Electronics Association (NMEA) standard GPS receiver (Furuno 1600F) and marine radar (Furuno 841).

Signal conditioning and data processing procedures for the collection of the auxiliary boat instrumentation data is included in the proposed Phase II program for the development and assessment of data to be recorded for post-mission analysis.

5.8 Portable Computer

A portable computer is recommended for use as a transportable data storage unit and for post-mission analysis of data. The HRA unit furnished under this Phase I program has a high-speed serial port and a data transfer sequence to allow transfer of the HRA recorded data to a portable computer.

6.0 INSTALLATION INSTRUCTIONS

The installation of the HRA system on either the MSW RIB or MK V SOC requires the installation of each component of the HRA at the designated location for that component. A recommended component, sensor, and cabling layout is presented in Figure 9. Sensing components must be positioned, oriented and fastened into position on the boat such that resultant data is properly correlated to known locations of the craft. After fastening each HRA component in its specified location, the components must be connected to the data acquisition component with the cables that are provided. The individual components of the HRA to be installed onto the craft consist of the following:

1. Data Acquisition Component: The Data Acquisition Component, shown in Figure 3, is a sealed enclosure approximately 12 x 8 x 8 inches (30.5 x 20.3 x 20.3 cm) and weighs 14 lbs (30.80 kg). This component contains the data acquisition and data processing computer and is expected to be mounted in the sub-console equipment area in any orientation. The component is required to be installed with sufficient clearance to allow access to the ten connector jacks on the upper face of the enclosure. This is necessary to provide clearance to connect the various sensors and HRA system components.
2. Console Display and Operator Input Component: The Console Display and Operators Input Component, shown in Figure 5, is a sealed enclosure approximately 8.5 x 4.75 x 2.5 inches (3.35 x 1.87 x 0.98 cm) and weighs 5 lbs (11kg). This component contains the display panel and the operator input switches. The component has a 26 conductor cable, eight feet long, that is easily connected, or disconnected, at the 26 pin console-

display bulkhead connector jack installed on the helmsman's console and as shown in Figure 6. This allows easy removal of this component when not in use. The component can be mounted to the helmsman's console at any preferred location utilizing the four ($\frac{1}{4}$ -20x $\frac{3}{4}$ ") tapped holes on the back of the enclosure.

3. Data Transfer and Display Connector Assembly: The Data Transfer and Display Connector Assembly, shown in Figure 6, contains a 10 pin connector jack with eight feet of electrical cable and a 26 pin connector jack with eight feet of cable. The two cables, one terminating with a 10 pin connector and one with a 26 pin connector, must be connected to the Data Acquisition Component. The 10 pin and 26 pin connector assembly is to be mounted at any convenient location on the console of the boat. This provides a data transfer connector and a display connector at the helmsman's console. It provides easy access for data transfer and connection of the display component at the console.
4. Triaxial Accelerometer Components (3 Units): The triaxial accelerometer components furnished under this program are environmentally sealed units. The units are each 4.5 by 2.25 by 1.25 inches (11.4 by 5.7 by 31.8 cm) and weigh 0.5 lb. (0.23 kg). The units have a (10) ten pin connector on the top surface as shown in Figure 7. The three units are to be mounted within the craft at three separate locations. The location and orientation of each triaxial accelerometer component are as follows:
 - Helmsman's Triaxial Accelerometer: This accelerometer is to be mounted under the floor at the helmsman's location as shown in Figure 9. The accelerometer must be mounted with the X-Y plane parallel to the deck. The X-axis must parallel the bow-stern centerline and point towards the bow. The Y-axis must lie pointed towards the port side. After installation, the triaxial accelerometer component must be connected to the Data Acquisition component with one of the 18 feet long, 10 conductor, cables furnished with the HRA.
 - Aft-port Triaxial Accelerometer: This accelerometer is to be mounted at the aft-port location shown in Figure 9. The accelerometer must be mounted with the X-Y plane parallel to the deck. The X-axis must parallel the bow-stern centerline and point towards the bow. The Y-axis must lie pointed towards the port side. After installation, the triaxial accelerometer component must be connected to the Data Acquisition component with one of the 35 feet long, 10 conductor, cables furnished with the HRA.
 - Aft-starboard Triaxial Accelerometer: This accelerometer is to be mounted at the aft-starboard location shown in Figure 9. The accelerometer must be mounted with the X-Y plane parallel to the deck. The X-axis must parallel the bow-stern centerline and point towards the bow. The Y-axis must lie pointed towards the port side. After installation, the triaxial accelerometer component must be connected to the Data Acquisition component with one of the 35 feet long, 10 conductor, cables furnished with the HRA.

5. Inclinometer and Electric Compass Component: The Inclinometer and Electronic Compass Component, shown in Figure 8, is housed in an environmentally sealed ABS plastic case that is 5 by 2.5 by 2.25 inches (12.7 by 6.4 by 5.7 cm) not including the integral connector jack on the (short) side and weighs 0.7 lb (1.5 kg). The unit has a (10) ten pin connector. The unit must be mounted with the X-Y plane parallel to the deck. The X-axis must parallel the bow-stern centerline and point towards the bow. The Y-axis must lie pointed towards the port side. The unit must be mounted in the subconsole area as specified in Figure 9. After installation, the component must be connected to the Data Acquisition component with the 12 foot long, 10 conductor cable furnished with the HRA.
6. Connectors and Cable Installation: The connectors and cables furnished under this Phase I program are presented in Table 1. All connectors must be securely locked to ensure their environmental seal. Figure 10 presents the available connectors associated with their respective HRA components. The cables should be run within conduits, either existing or new, below the deck without interfering with any other systems. The +24 VDC supply cable is not provided under this Phase I program, therefore it must be supplied by the boat manufacturer.

Table 1. Connector/Cable Specifications

Connector	Type	# of Pins	Description
J1	Circular Bendix	26	Display Control at Data Acquisition Component
J2	Circular Bendix	26	Display Control at Bulkhead Connector
J3	Circular Bendix	10	Aft-Port Side Accelerometer
J4	Circular Bendix	10	Aft-Starboard Side Accelerometer
J5	Circular Bendix	10	Helmsman Accelerometer
J6	Circular Bendix	10	Helmsman Accel. at Data Acquisition Component
J7	Circular Bendix	10	Aft-Starboard Accel. at Data Acquisition Component
J8	Circular Bendix	10	Aft-Port Accel. at Data Acquisition Component
J9	Circular Bendix	10	Compass/Horizon at Data Acquisition Component
J10	Circular Bendix	10	Electronic Compass/Horizon Sensor
J11	Circular Bendix	4	Ship Power Connection at Data Acquisition Component
J12	Circular Bendix	26	Parallel Input/Output at Data Acquisition Component
J13	Circular Bendix	10	Serial Input/Output at Data Acquisition Component
J14	Circular Bendix	10	Serial Input/Output at Bulkhead Connector
J15	Circular Bendix	6	Spare Connector
J16	Circular Bendix	10	Spare Connector

Furnished Cable	From Connector	To Connector	Available Length (feet)	Notes
C1	J1	J2	12	
C2	J2	Display	8	Hardwired at Display
C3	J9	J10	12	
C4	J8	J3	30	
C5	J7	J4	30	
C6	J6	J5	15	
C7	J13	J14	12	

7.0 OPERATING INSTRUCTIONS

The application software is written in Microsoft C v5.1 and Microsoft MASM v5.0 as a real-time, embedded program that executes in real mode on an Intel 486DX-100 microprocessor. A user interface is provided by the display and four environmentally sealed, momentary contact, pushbuttons. The functions of the pushbuttons are indicated by a programmable menu on the display immediately adjacent and to the left of the pushbuttons.

The display initially shows two vertical bar graphs (see Figure 11) which are the aft and helm absolute (unsigned) vertical *peak instantaneous* accelerations with an initial scale of 5g. The scale can be toggled between 5g and 10g full scale. The aft acceleration displayed is the maximum of either the aft port or aft starboard vertical accelerations. The two triangular marks next to each of the vertical bars will show the *moving average* of the peak aft and helm acceleration. The *moving average interval* is system (not user) programmable and initially set to 4 seconds. The display also shows RECORD OFF, indicating that the three triaxial accelerations, compass, pitch and roll data are not being recorded.

The first pushbutton on the initial menu bar is AZM/HD (azimuth/heading), which will change the display to one which shows the pitch and roll of the vessel, the electronic compass heading and a second menu (see Figure 12). The second pushbutton is SCALE, which changes the gain of the acceleration vertical bar graph. Although any gain can be programmed, SCALE toggles the display between 5g and 10g full scale. The third pushbutton is unused and displays a blank. The fourth pushbutton, labeled SYSTEM, clears the display and presents another menu (see Figure 13).

If AZM/HD is pushed, the display changes to that shown in Figure 12. The *artificial horizon indicator* (pitch and roll) is shown on the left and the *compass* on the right. These auxiliary transit measurements are recorded and used in the post-mission analysis of the performance of the helmsman. The first pushbutton in this menu is ACCEL (accelerometers), which will change the display back to that shown in Figure 11. The second and third pushbuttons are unused and display a blank. The fourth pushbutton is SYSTEM, which clears the displays and presents a third menu (see Figure 13).

The first pushbutton of the SYSTEM menu is START RECORD, which produces another menu (see Figure 14) to initiate data recording. The second pushbutton is START DATA, which produces another menu (see Figure 16) to initiate the transfer of stored data to a remote computer. The third pushbutton is SETUP, which produces another menu (see Figure 18) for the diagnostic maintenance of the HRA. The fourth pushbutton is EXIT, which returns the display back to that shown in Figure 11.

The first pushbutton of the RECORD menu (see Figure 14) is RECORD UNCOMP (record uncompressed), which will store the data in an uncompressed format (see Appendix B: Data Format and Transfer). The second pushbutton is RECORD COMP (record compressed) which will store the triaxial acceleration data as the difference between sample values (see Appendix B:

Data Format and Transfer) to conserve data storage. The third pushbutton is unused and displays a blank. The fourth pushbutton is EXIT, which returns the display back to that shown in Figure 11.

If either the RECORD UNCOMP or the RECORD COMP pushbutton was depressed, then the initial display shows RECORD ON indicating that the three triaxial accelerations, compass, pitch and roll data are being recorded.. The first pushbutton of this first menu is now STOP RECORD, which will terminate data recording and close the storage data file (see Appendix B: Data Format and Transfer) as shown in Figure 15.

The first pushbutton of the DATA transfer menu (see Figure 16) is START DATA, which initiates a data transfer. The second and third pushbuttons are unused and display a blank. The fourth pushbutton is EXIT, which returns the display back to that shown in Figure 11. Data transfer should not be initiated without a remote data communication session (see Appendix B: Data Format and Transfer). If this occurs, either a remote session should be started or the HRA should be reinitiated (+24 VDC power down and power up).

If the data transfer is initiated the display shows a prompt without a menu, as shown in Figure 17.

If the SETUP pushbutton is pushed in the SYSTEM menu (see Figure 13), the display changes to that shown in Figure 18. The first pushbutton of the SETUP, or diagnostic, menu is ACCEL (accelerometers). If ACCEL is pushed, the display changes to that shown in Figure 19 which shows a maintenance numerical display of the three triaxial accelerometers. The second pushbutton is DISPLY (display). If DISPLY is pushed, the display changes to that shown in Figure 20 which maintains the clarity of the LCD. The third pushbutton is unused and displays a blank. The fourth pushbutton is EXIT, which returns the display back to that in Figure 11.

If the ACCEL pushbutton is pushed in the SETUP menu, the display changes to that shown in Figure 19. A maintenance display of the three triaxial accelerometer sensors is shown in instantaneous, calibrated (G) numerical form to verify the integrity of the nine sensors. The first, second and third pushbuttons are unused and display a blank. The fourth pushbutton is EXIT, which returns the display back to that in Figure 11.

If the DISPLY (display) pushbutton is pushed in the SETUP menu, the display changes to that shown in Figure 20. The first pushbutton of the DISPLY menu is ADJUST, which varies the contrast of the LCD linearly from its normal (midscale) value to low contrast, then high contrast, and back through low contrast continuously. The second pushbutton is CLEAR, which cycles automatically and clears and restores contrast to the LCD. The third pushbutton is unused and displays a blank. The fourth pushbutton is EXIT, which maintains any contrast adjustments made and returns the display back to that in Figure 11.

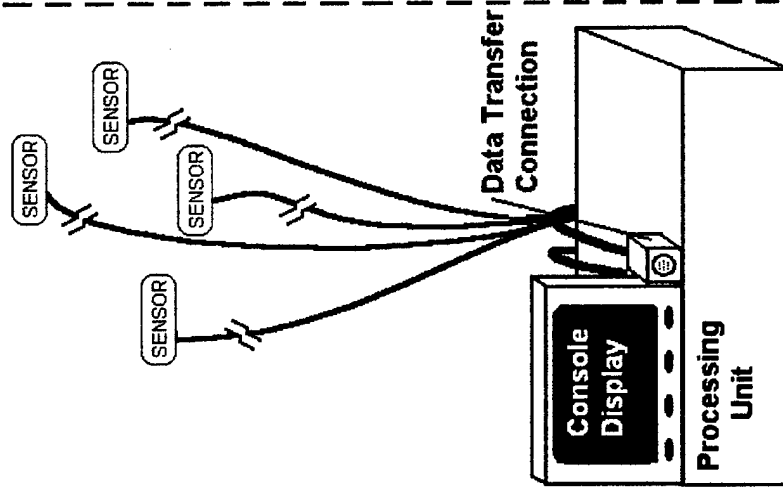
8.0 REFERENCES

- ISO 2631, "Guide to the Evaluation of Human Exposure to Whole Body Vibration – Part 1: General Requirements," ISO/DIS 2631-1, International Organization for Standardization, Geneva, 1994.
- Giles, David L., "Faster Ships for the Future: New Designs for Oceangoing Freighters May Soon Double their Speed," *Scientific American*, p. 126, October 1997.
- Gillmer, Thomas C., "Modern Ship Design," Second Edition, U.S. Naval Institute Press, 1986.
- Lesh, Donald B., "Seakeeping Characteristics of Slice Hulls: A Motion Study in Six Degrees of Freedom," NTIS Report No. ADA304157, p. 58, 1995.
- McCreight, K. K., "Assessing the Seaworthiness of SWATH Ships," *SNAME Transactions*, vol. 95, pp. 189-214, 1987.
- Silage, D., Hartmann, B., Quartuccio, J., and Nichols, J., "Pelvis Mounted Data Acquisition System and Novel Transducers for a Fifth Percentile Female Manikin," *Proceedings 33rd SAFE Conference*, pp. 210-216, 1995.

APPENDIX A

FIGURES

HRA Design



Data Transfer and Post-Mission Analysis System

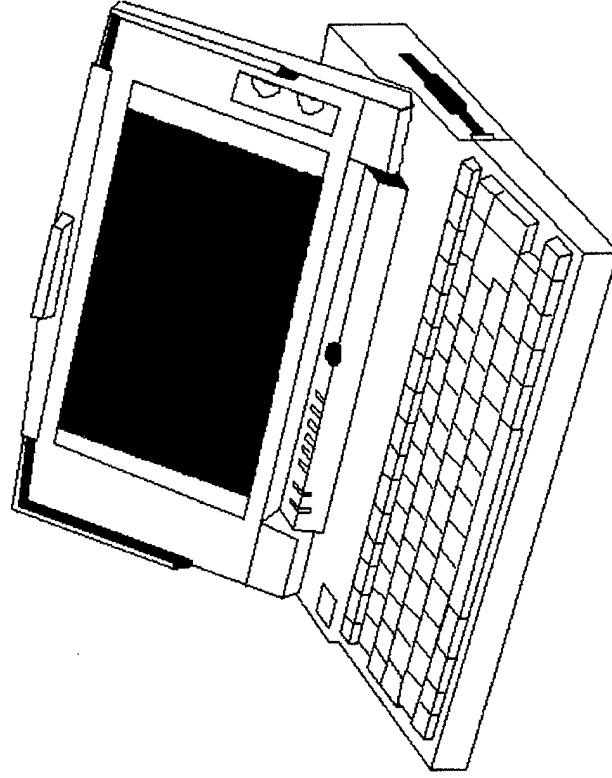


Figure 1. HRA General Arrangement

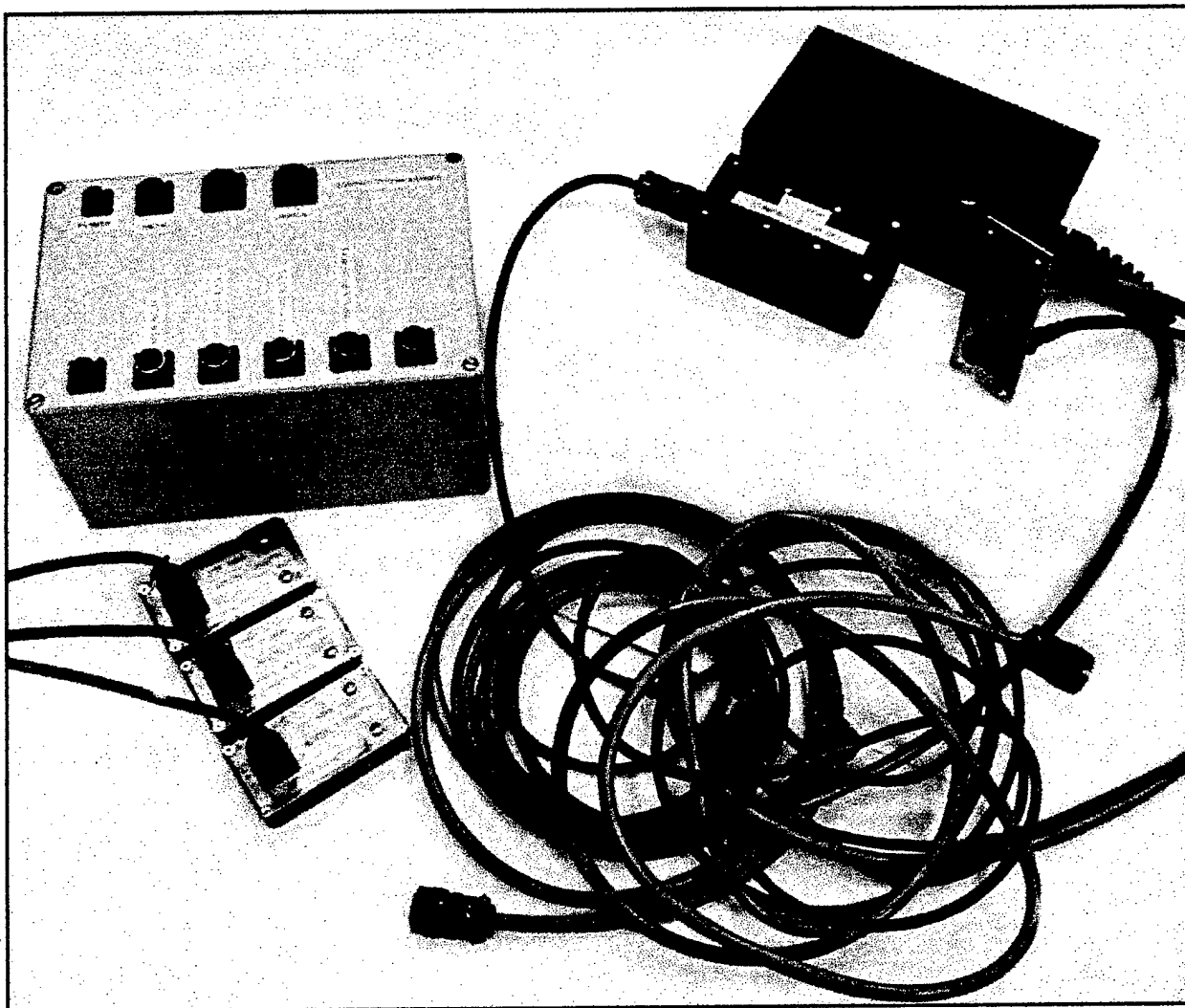


Figure 2. Helmsman Recording Accelerometer Components

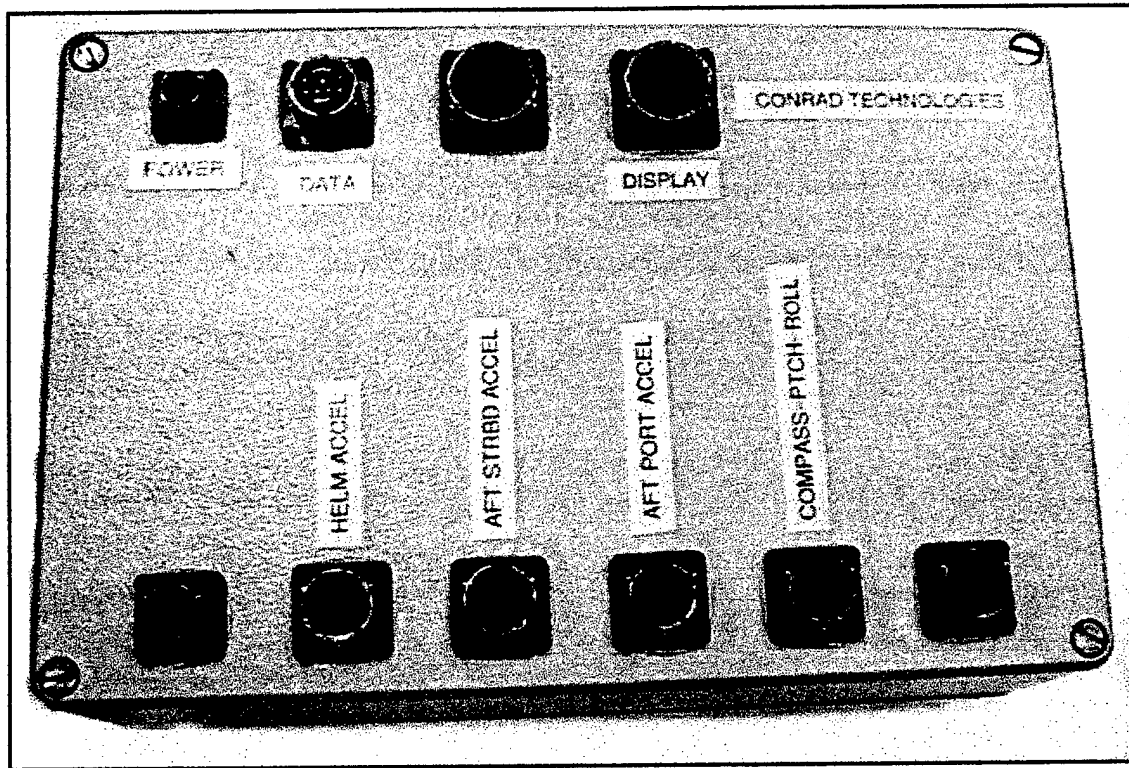


Figure 3. Data Acquisition Component

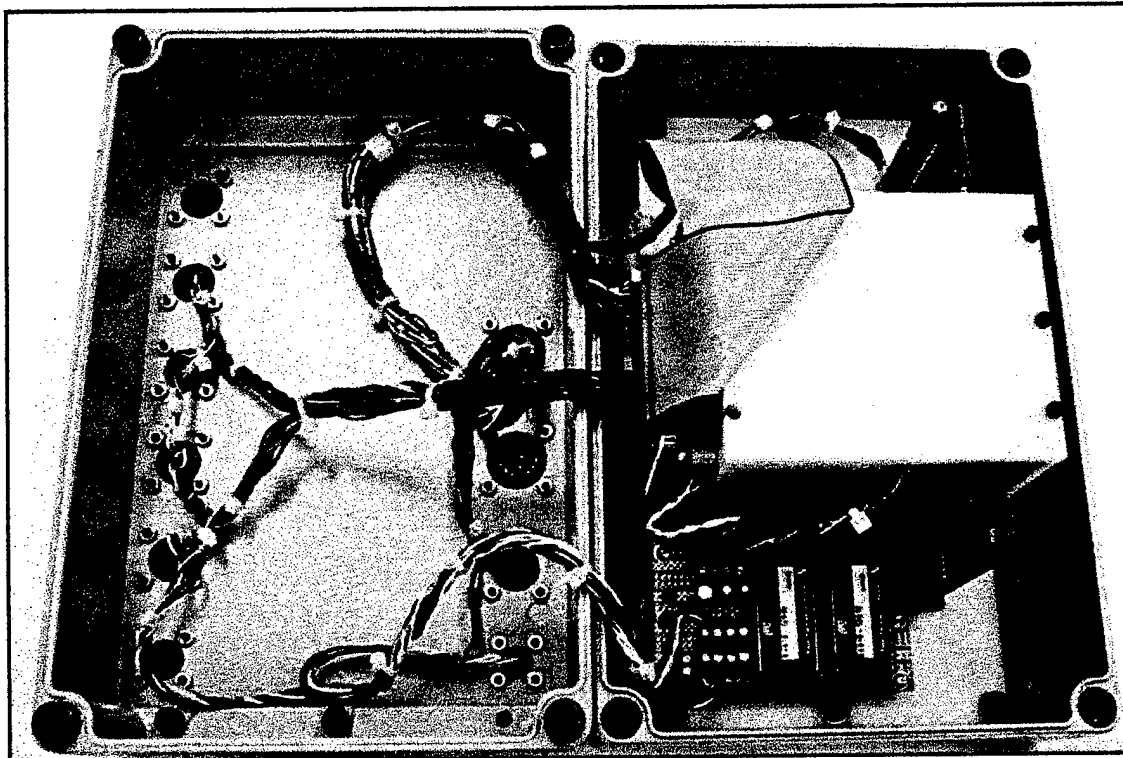


Figure 4. Data Acquisition and Processing Electronics

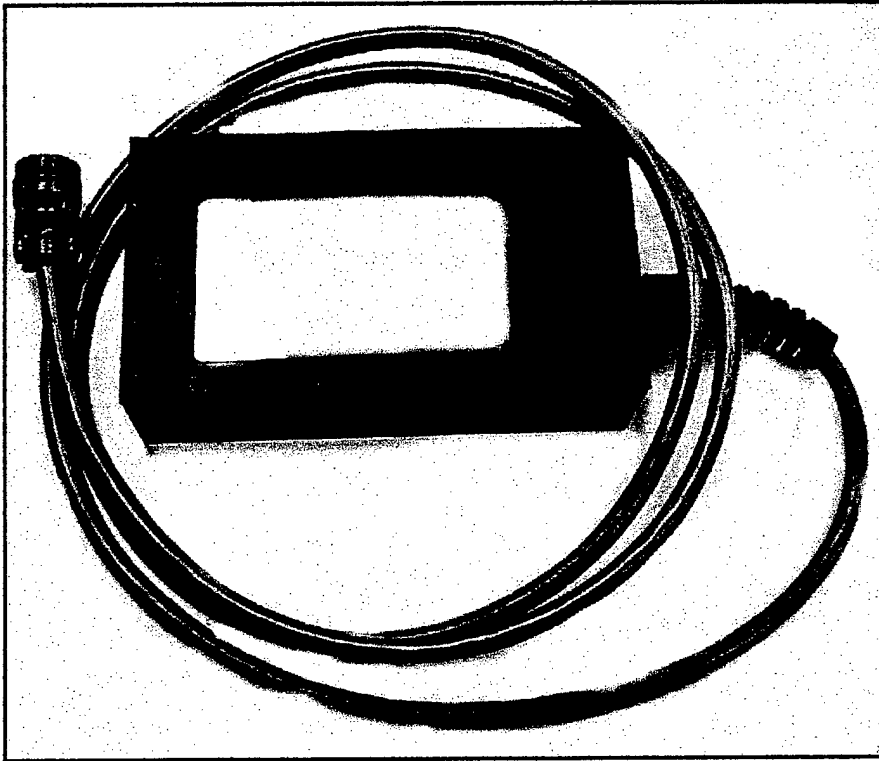


Figure 5. Console Display and Operator's Input Component

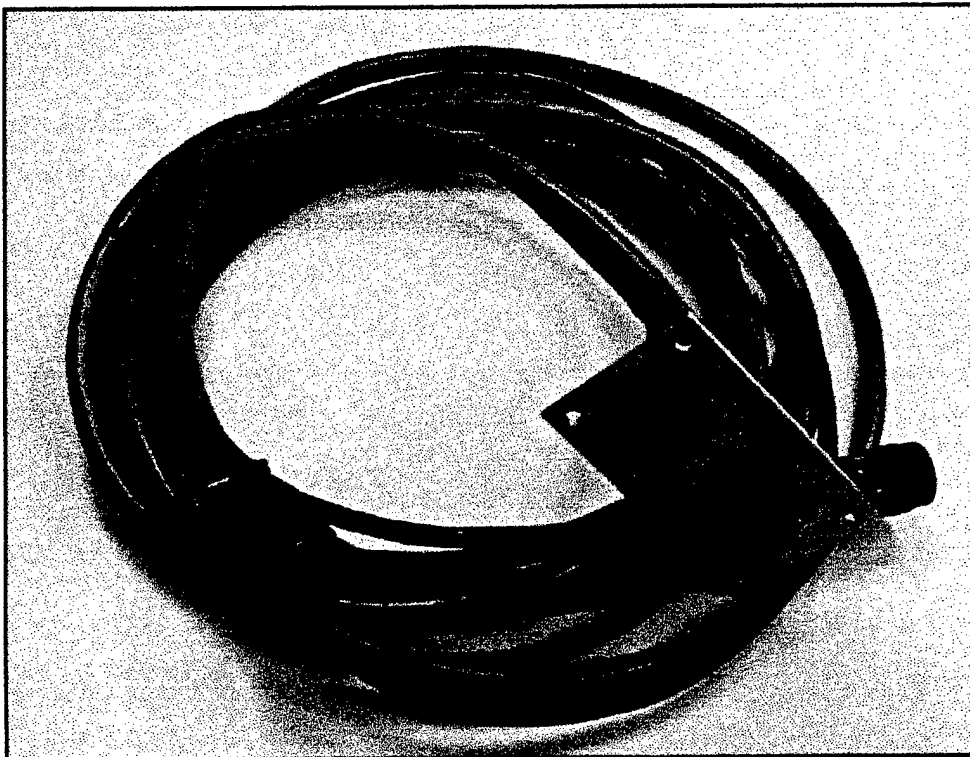


Figure 6. Data Transfer and Display Connector Assembly

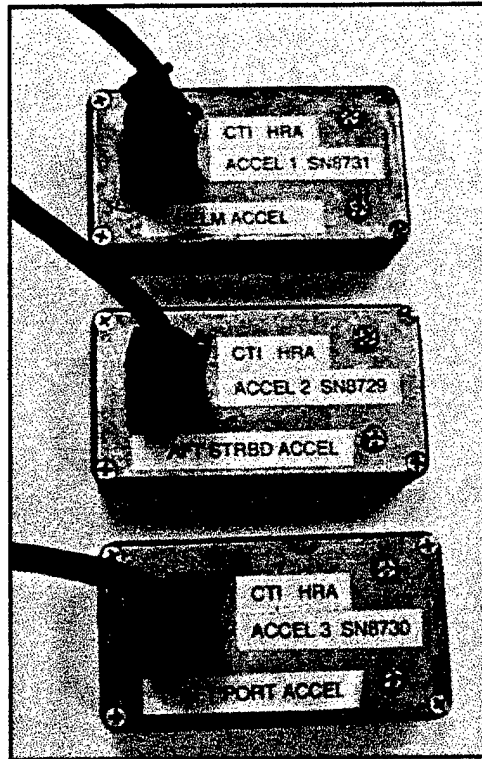


Figure 7. Triaxial Accelerometer Component

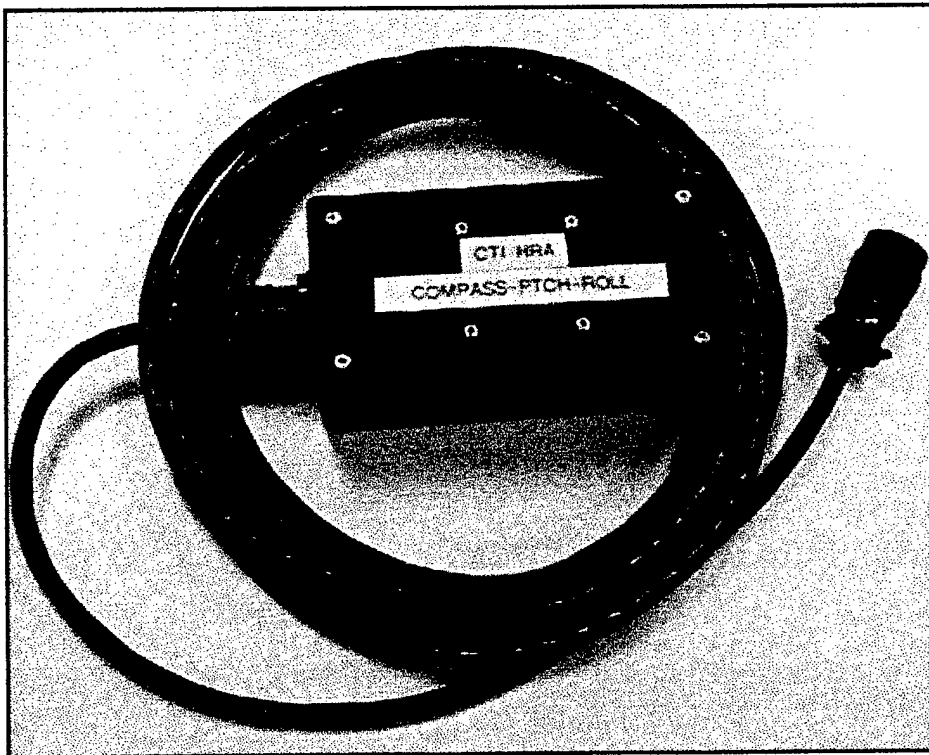


Figure 8. Inclinometer and Electronic Compass Component

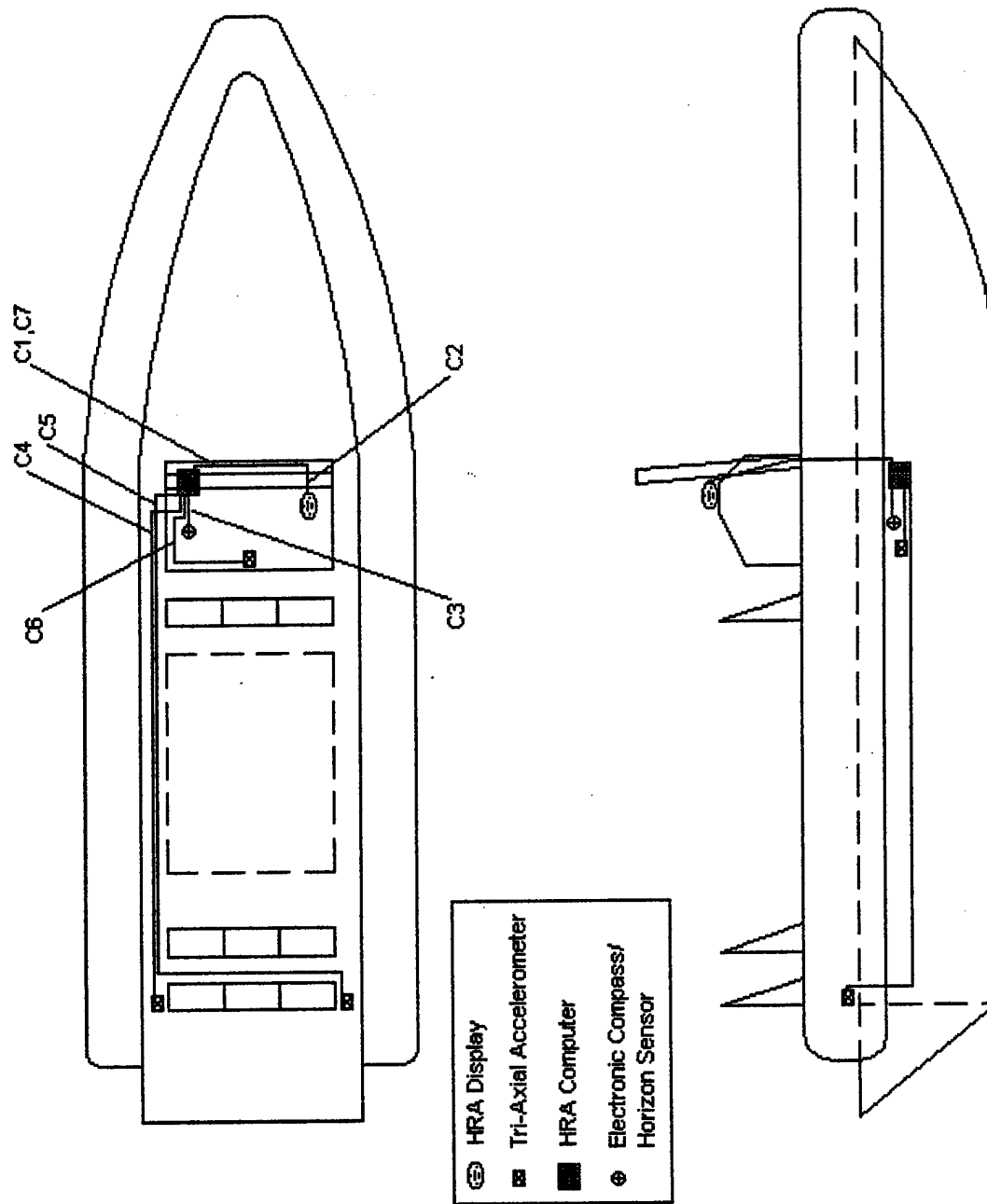


Figure 9. Component, Sensor, and Cabling Layout

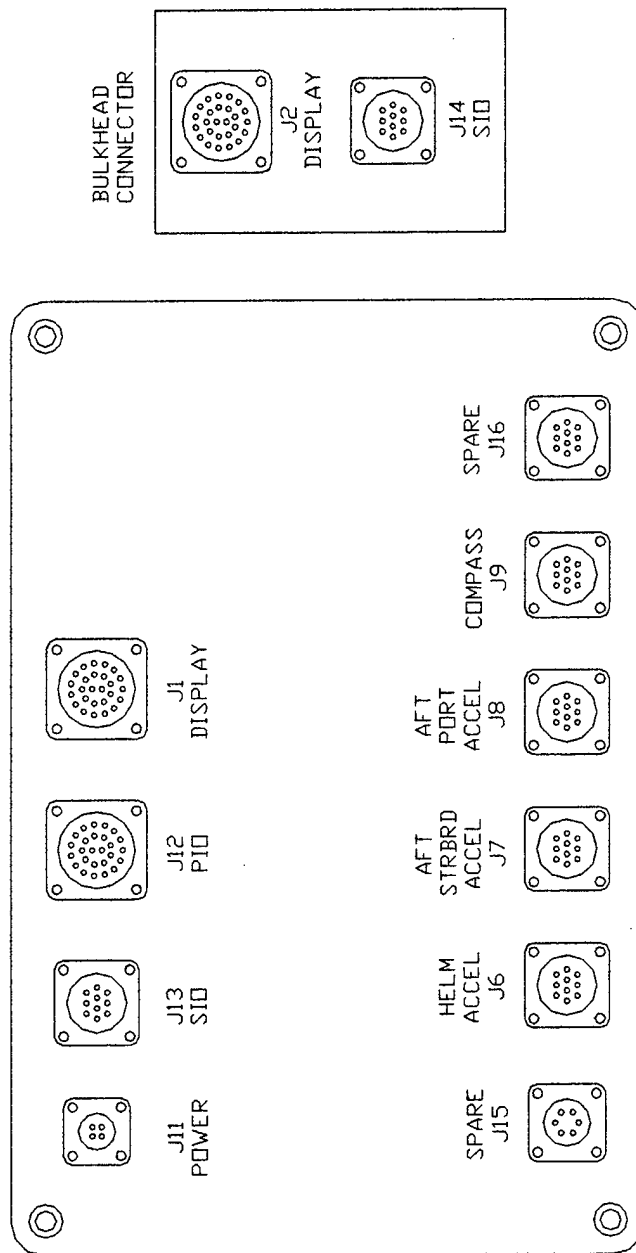
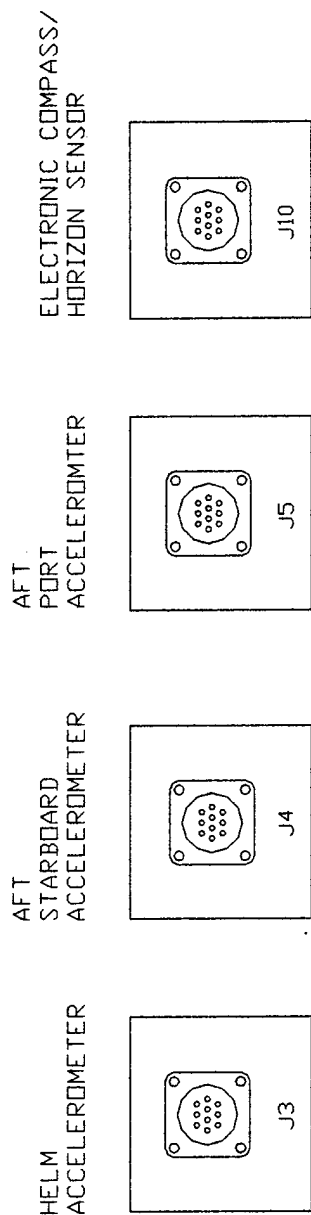


Figure 10. Connector Layout

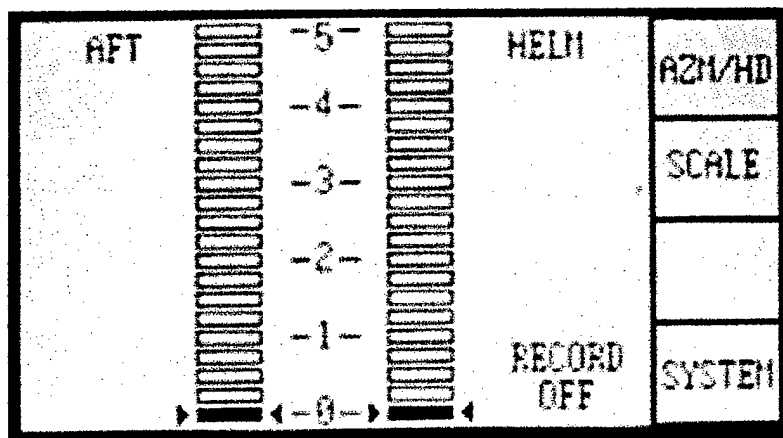


Figure 11. Aft and helm peak and moving average vertical acceleration display.

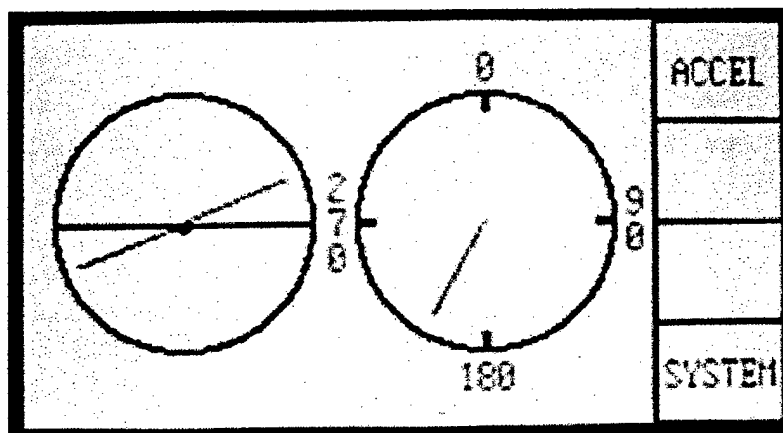


Figure 12. Artificial horizon and compass data display.

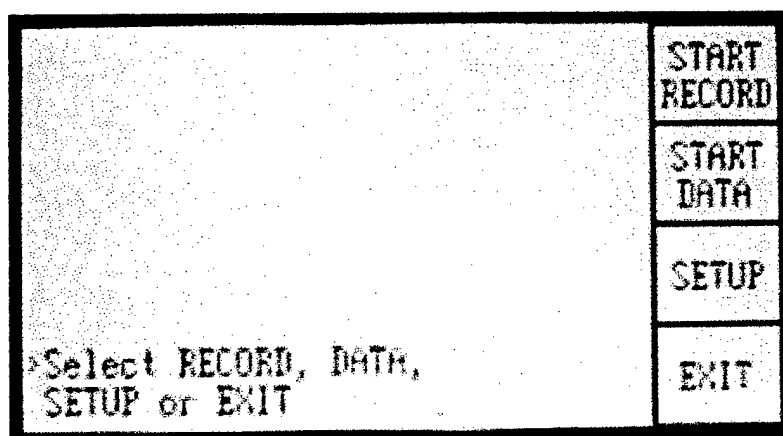


Figure 13. SYSTEM menu.

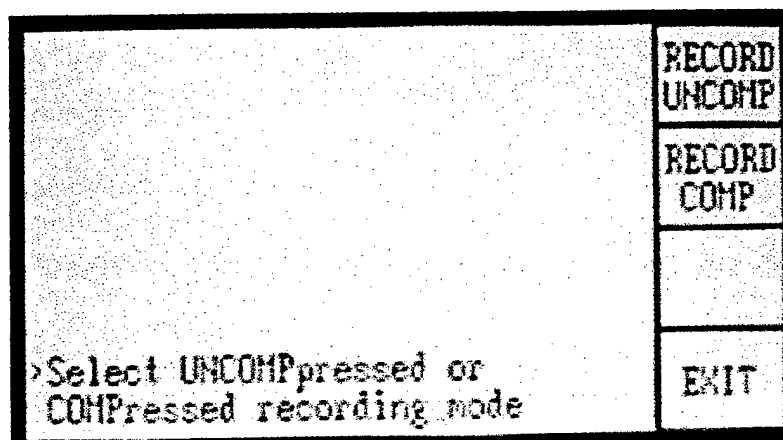


Figure 14. RECORD menu.

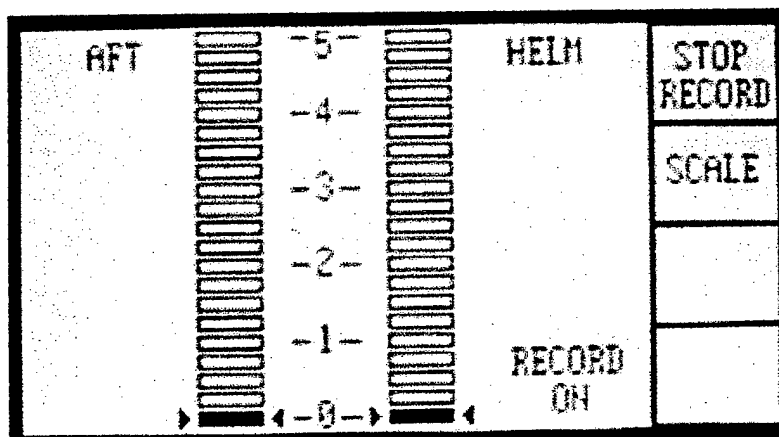


Figure 15. Aft and helm acceleration display during RECORD mode.

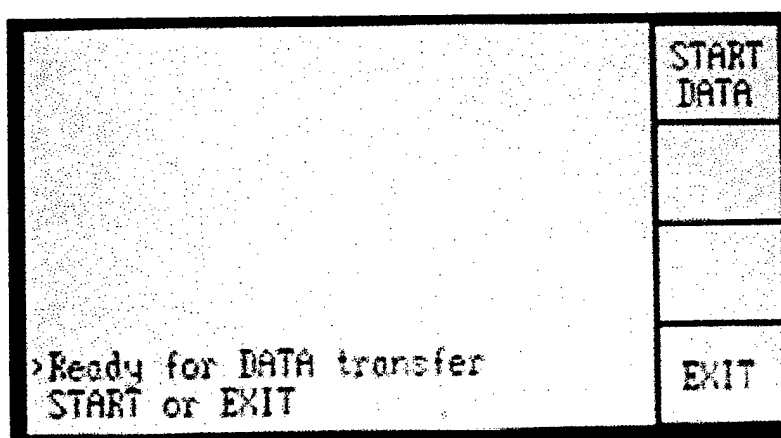


Figure 16. DATA transfer menu.

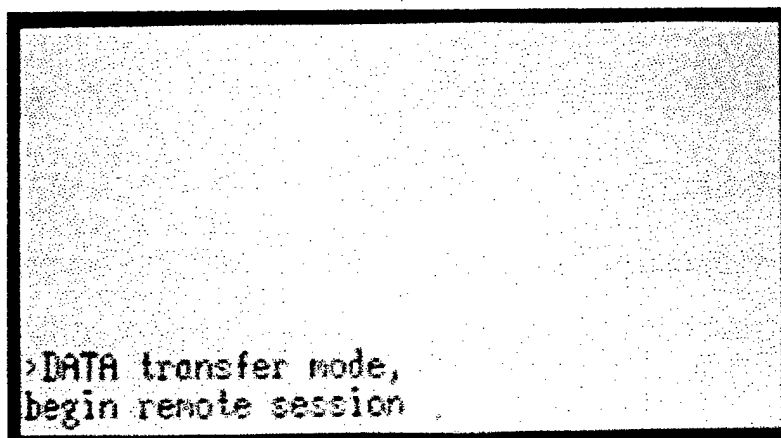


Figure 17. Display during DATA transfer.

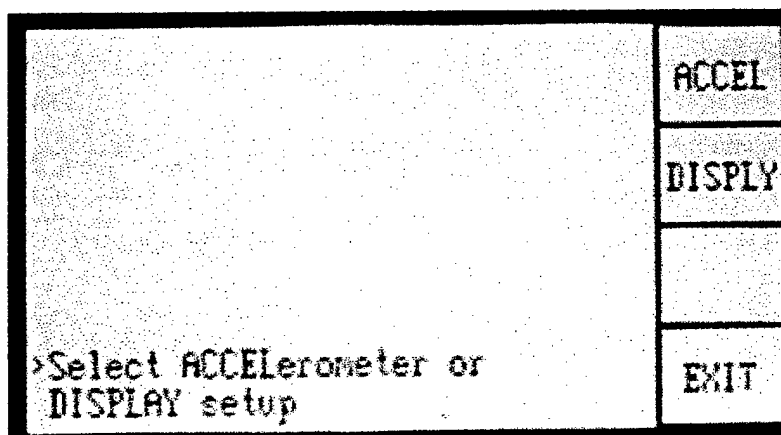


Figure 18. SETUP (diagnostic) menu.

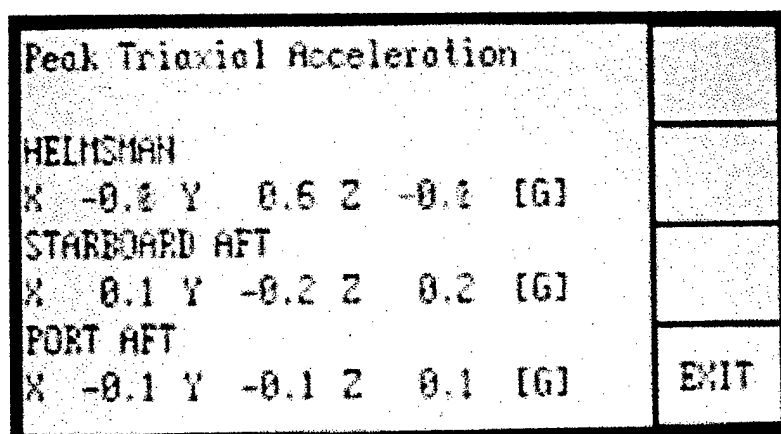


Figure 19. Peak triaxial acceleration (ACCEL) diagnostic display

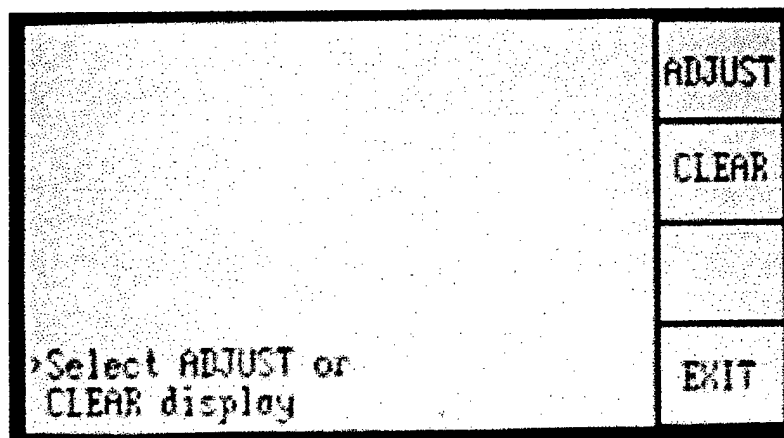


Figure 20. DISPLY (display) diagnostic menu.

APPENDIX B

DATA FORMAT AND TRANSFER

The HRA sensor calibration is an *ASCII file* entitled *helmcal.dat*. The calibration file structure is as follows, with U indicating computer units:

integer	diagnostic parameter (usually set to 0 for normal operation)
integer	helm X acceleration zero [U]
floating point	helm X acceleration calibration [U/G]
integer	helm Y acceleration zero [U]
floating point	helm Y acceleration calibration [U/G]
integer	helm Z acceleration zero [U]
floating point	helm Z acceleration calibration [U/G]
integer	aft starboard X acceleration zero [U]
floating point	aft starboard X acceleration calibration [U/G]
integer	aft starboard Y acceleration zero [U]
floating point	aft starboard Y acceleration calibration [U/G]
integer	aft starboard Z acceleration zero [U]
floating point	aft starboard Z acceleration calibration [U/G]
integer	port starboard X acceleration zero [U]
floating point	port starboard X acceleration calibration [U/G]
integer	port starboard Y acceleration zero [U]
floating point	port starboard Y acceleration calibration [U/G]
integer	port starboard Z acceleration zero [U]
floating point	port starboard Z acceleration calibration [U/G]
integer	pitch zero
integer	roll zero

The data file is a *binary file* entitled *mmddhhnn.hra*, where the file name is constructed from the two digit *month (mm)*, *day (dd)*, *hour (hh)*, and *minute (nn)* when the file was opened. This construct gives a unique file name that identifies the information stored by the date and time of its inception. The data file structure is as follows:

integer	compression (1 compressed, 0 uncompressed data)
integer	helm X acceleration zero [U]
integer	helm X acceleration calibration [U/G]
integer	helm Y acceleration zero [U]
integer	helm Y acceleration calibration [U/G]
integer	helm Z acceleration zero [U]
integer	helm Z acceleration calibration [U/G]
integer	aft starboard X acceleration zero [U]
integer	aft starboard X acceleration calibration [U/G]
integer	aft starboard Y acceleration zero [U]
integer	aft starboard Y acceleration calibration [U/G]
integer	aft starboard Z acceleration zero [U]
integer	aft starboard Z acceleration calibration [U/G]
integer	port starboard X acceleration zero [U]
integer	port starboard X acceleration calibration [U/G]

integer	port starboard Y acceleration zero [U]
integer	port starboard Y acceleration calibration [U/G]
integer	port starboard Z acceleration zero [U]
integer	port starboard Z acceleration calibration [U/G]
integer	pitch zero
integer	roll zero

This is followed by the nine channels of acceleration data stored either in *byte* or *integer* format as dictated by the compression parameter. The next three *integers* (regardless of compression) is the compass heading in U with a scale factor of 11.37 U/degree and the pitch and roll both in U and both with a scale factor of 34.13 U/degree. This set of 12 measurements is then repeated at 50 samples per second until recording is manually stopped (see Figure 14) or the non-volatile storage medium is full and the recording is automatically terminated.

There is a minimum of 22 MB of non-volatile data storage available, which provides over 8 hours of recording time in compressed mode and over 5 hours in uncompressed mode. All data files are common MS DOS files, share the available storage and would be normally transferred from the HRA to a remote (laptop) computer for further processing and analysis .

Data transfer is started from the SYSTEM selection on the initial acceleration display (see Figure 11). SYSTEM is only available if data recording is off (RECORD OFF shown, as in Figure 11). The SYSTEM menu (see Figure 13) has the START DATA selection available which produces the DATA transfer menu (see Figure 16). This evokes an MS DOS mode session of pcANYWHERE for DOS (Symantec) to begin. The data transfer is through a 10 conductor cable to a high speed serial port of an external computer executing pcANWHERE 32 (Symantec, Windows 95/Windows NT) version 8. The initialized communication parameters are 8 bit, no parity, at 19.2 Kb/sec.

The remote session allows full, external control of the embedded HRA processing unit. Data files can be transferred and deleted, the calibration file can be edited, and the application software (*helm.exe*) can be edited, compiled, and linked and downloaded to the HRA for further performance and operational upgrades.

APPENDIX C

SOURCE CODE

The C language source code for the embedded application software (*helm.c*) and the 8086 assembly language subroutines (*helmsub.asm*) are listed below.

```

1      /* Helmsman Recording Accelerometer
2      c1998 Conrad Technologies, Paoli PA

3      helm.c

4      FLASH storage 22 MB (27 MB formatted)
5      compass 11.37 U/deg iacc[10] 0-359 deg
6      pitch 34.13 U/deg iacc[11] +-60 deg
7      roll 34.13 U/deg iacc[12] +-60 deg
8      */

9      #include <stdio.h>
10     #include <io.h>
11     #include <fcntl.h>
12     #include <graph.h>
13     #include <dos.h>
14     #include <string.h>
15     #include <bios.h>
16     #include <math.h>
17     #include <time.h>

18     char    string[40] = "static string";
19     char    shdazm[7]="AZM/HD";
20     char    saccel[6]="ACCEL";
21     char    sdisply[7]="DISPLY";
22     char    srecord[7]="RECORD";
23     char    suncomp[7]="UNCOMP";
24     char    scomp[5]="COMP";
25     char    sdata[5]="DATA";
26     char    son[3]="ON";
27     char    soff[4]="OFF";
28     char    sadjust[7]="ADJUST";
29     char    sclear[6]="CLEAR";
30     char    scntrst[7]="CNTRST";
31     char    sreset[6]="RESET";
32     char    sstart[6]="START";
33     char    sstop[5]="STOP";
34     char    ssystem[7]="SYSTEM";
35     char    sscale[6]="SCALE";
36     char    ssetup[6]="SETUP";
37     char    smon[4]="MON";
38     char    sexit[5]="EXIT";
39     char    s180[4]="180";

40     char    fbuffer[2000];
41     char    sdate[12],stime[12],sfile[20];
42     char    bdata;

43     int     i,j,irecord,ipb;
44     int     ifhandle,idiagn;

```

```

45     int      icomp,ichan,idata,ilcd;
46     int      iacc[16],ilacc[16],ichz[16],ichg[16];
47     int      iaccgn,ivaca,ivaccla,ivach,ivachh;
48     int      ivacmap,ivacama,ivacamas[50],ivachma;
49     static int ivachmas[50];
50     int      ivacmapmx,ivacptc,ivacap,ivacaptc,ivachp,ivachptc;
51     int      ihdazmd,icmps,icmpl,iptch,iptchl,iptchz,irot,irotl,irotz;
52     int      ix,iy,ixl,iyl;

53     long      lrecnum;

54     float      ftemp;
55     float      fchg[16];

56     FILE      *fp;

57     main()
58     {

59         /* initialization and start */

60         start:    inithwd();          /* initialize hardware */
61         lcdoff();      /* LCD bias off */
62         dispon();      /* display ON */
63         clear();       /* clear LCD */
64         for (iy=0;iy<=127;iy++)
65         {
66             for (ix=0;ix<=239;ix++)
67                 setpix(ix,iy);
68             for (ix=0;ix<=239;ix++)
69                 clrpx(ix,iy);
70         }
71         lcdmid();      /* LCD bias midscale */
72         ilcd=32;

73         /* helmcad.dat diagnostic (0/1)
74         helmsman X Y Z aft starboard X Y Z aft port X Y X
75         accelerometer calibration: ichz zero U, fchg gain U/G
76         pitch zero, roll zero */

77         if ((fp=fopen("helmcad.dat","r"))==NULL)
78             goto erraccel;
79         fscanf(fp,"%d\n",&idiagn); /* diagnostic */
80         for (i=1;i<=9;i++)
81         {
82             fscanf(fp,"%d\n%f\n",&ichz[i],&fchg[i]);
83             ichg[i]=fchg[i]+0.5;
84         }
85         fscanf(fp,"%d\n%d\n",&iptchz,&irotz);
86         fclose(fp);

87         /* open data file mmdhmm.hra
88         compression (0/1), 9[accel channel zero U, accel channel gain fp U/G]
89         pitch zero U, roll zero U, {9[accel data (byte/word) U],

```

```

90      compass U, pitch U, roll U}.....*/

91      restart:      _strdate(sdate);
92      _strtime(stime);
93      i=0;
94      j=0;
95      sfscca:      if (sdate[i] == '/')
96      goto sfscsca;
97      sfile[j]=sdate[i];
98      i++;
99      j++;
100     goto sfscsca;
101     sfscsca:      i++;
102     sfscsb:      if (sdate[i] == '/')
103     goto sfscsb;
104     sfile[j]=sdate[i];
105     i++;
106     j++;
107     goto sfscsb;
108     sfscsb:      i=0;
109     sfscsc:      if (stime[i] == ':')
110     goto sfscsc;
111     sfile[j]=stime[i];
112     i++;
113     j++;
114     goto sfscsc;
115     sfscsc:      i++;
116     sfscsd:      if (stime[i] == ':')
117     goto sfscsd;
118     sfile[j]=stime[i];
119     i++;
120     j++;
121     goto sfscsd;
122     sfscsd:      sfile[j]='.';
123     j++;
124     sfile[j]='h';
125     j++;
126     sfile[j]='r';
127     j++;
128     sfile[j]='a';
129     j++;
130     sfile[j]='\0';
131     ifhandle=open(sfile, O_BINARY | O_WRONLY | O_CREAT);
132     if (ifhandle==-1)
133     goto errfopen;

134     /* initialize parameters */

135     icalp=0;          /* uncompressed data */
136     iaccgn=2;         /* accelerometer display gain */
137     irecord=0;        /* recorder OFF */
138     ivacmapmx=50;     /* moving average size */
139     ivacptc=200;      /* peak acceleration TC */

```

```

140      /* banner */

141      clear();
142      strcpy(string,"HELMSMAN ACCELERATION DISPLAY v1.2");
143      lcdtext(20,65,string);
144      strcpy(string,"Conrad Technologies");
145      lcdtext(60,45,string);
146      rectangle(10,38,220,50);
147      rectangle(9,37,222,52);
148      wait(1500);

149      /* acceleration display setup */

150      accdsps:   clear();
151      menu();
152      waitpb();
153      lcdtext(203,72,sscale);
154      lcdtext(153,12,srecord);
155      if (irecord==0)
156      {
157          lcdtext(201,102,shdazm);
158          lcdtext(201,6,ssystem);
159          lcdtext(162,0,soff);
160      }
161      else
162      {
163          lcdtext(206,108,sstop);
164          lcdtext(201,96,srecord);
165          lcdtext(166,0,son);
166      }
167      dispacc(iaccgn);
168      ivacla=0;          /* last vertical acceleration */
169      ivaclh=0;
170      ivacap=0;          /* peak vertical acceleration */
171      ivachp=0;
172      ivacaptc=ivacptc;  /* peak vertical acceleration TC */
173      ivachptc=ivacptc;
174      for (i=1;i<=ivacmapmx;i++) /* zero moving average storage */
175      {
176          ivacamas[i]=0;
177          ivachmas[i]=0;
178      }
179      ivacmap=1;          /* moving average pointer */
180      ivacama=0;          /* moving average value */
181      ivachma=0;

182      /* acceleration display */

183      accdsp:   readpb(&ipb);      /* 0 none
184      hd/azm display
185      accelerometer display gain
186      exit on diagnostic
187      system */
188      if (idiagn==0)

```

```

189      goto accdpb;
190      if (ipb==3)
191      goto extpgm;

192      accdpb:      if (ipb==2)          /* accelerometer gain */
193      goto accgain;

194      if (irecord==1)
195      goto accdspr;

196      if (ipb==1)          /* only if not recording */
197      goto hdazmdsps;
198      if (ipb==4)
199      goto sysfunct;
200      goto accdspc;

201      accdspr:      if (ipb==1)          /* stop recording */
202      goto stoprec;

203      accdspc:      wclock();          /* data sampling */
204      for (i=1;i<=6;i++)
205      adc1(i,&iacc[i]);
206      for (i=1;i<=6;i++)
207      adc2(i,&iacc[i+6]);

208      if (irecord==0)
209      goto accdcal;

210      if (lrecnum==0)      /* recording data */
211      {
212      for (i=1;i<=12;i++)
213      ilacc[i]=iacc[i];
214      }
215      if (icompr==0)
216      goto recuncomp;

217      j=0;          /* compressed recording */
218      for (i=1;i<=9;i++)      /* delta accel */
219      {
220      idata=iacc[i]-ilacc[i];
221      ilacc[i]=iacc[i];
222      bdata=idata;
223      fbuffer[j]=bdata;
224      j++;
225      }
226      for (i=10;i<=12;i++)      /* compass, pitch, roll */
227      {
228      fbuffer[j]=iacc[i];
229      j++;
230      j++;
231      }
232      if(write(ifhandle, fbuffer, 15)!=15) /* data recording */
233      goto recerr;
234      lrecnum++;

```

```

235      if (lrecnum<=1466666) /* 15 x 1.4666 MB = 22 MB */
236      goto accdcal;
237      goto stoprec;

238      recuncomp:    j=0;                /* uncompressed recording */
239      for (i=1;i<=12;i++)
240      {
241      fbuffer[j]=iacc[i];
242      j++;
243      j++;
244      }
245      if(write(ifhandle, fbuffer, 24)!=24) /* data recording */
246      goto recerr;
247      lrecnum++;
248      if (lrecnum<=916666) /* 24 x 0.91666 MB = 22 MB */
249      goto accdcal;
250      goto stoprec;

251      accdcal:      idata=iaccgn*(abs(iacc[3]-ichz[3])/ichg[3]); /* HELM Az */
252      ivacmap++;    /* check moving average pointer */
253      if (ivacmap>ivacmapmx)
254      ivacmap=1;
255      ivachma=ivachma+idata-ivachmas[ivacmap];
256      ivachmas[ivacmap]=idata;
257      ivach=ivachma/ivacmapmx;
258      if(ivach>=20) /* maximum display value */
259      ivach=20;

260      if (ivach>ivachp) /* check peak value */
261      {
262      clrthres(ivachp,1);
263      ivachp=ivach;
264      ivachptc=ivacptc;
265      setthres(ivachp,1);
266      }
267      if (ivach==ivaclh)
268      goto acchpdsp;
269      if (ivach>ivaclh) /* HELM acceleration display */
270      {
271      for (i=ivaclh+1;i<=ivach;i++)
272      setvacc(i,1);
273      }
274      if (ivach<ivaclh)
275      {
276      for (i=ivaclh;i>ivach;i--)
277      clrvacc(i,1);
278      }
279      ivaclh=ivach;

280      acchpdsp:    if (ivachp<=0) /* peak HELM acceleration display */
281      goto accadsp;
282      ivachptc--;
283      if (ivachptc==0)
284      {

```

```

285     ivachptc=ivacptc;
286     clrthres(ivachp,1);
287     ivachp--;
288     setthres(ivachp,1);
289     }

290     accadsp:    if (iacc[6]>=iacc[9]) /* select greatest AFT Az */
291     ichan=6;
292     else
293     ichan=9;
294     idata=iaccgn*(abs(iacc[ichan]-ichz[ichan])/ichg[ichan]);
295     ivacama=ivacama+idata-ivacamas[ivacmap];
296     ivacamas[ivacmap]=idata;
297     ivaca=ivacama/ivacmapmx;
298     if(ivaca>=20)      /* maximum display value */
299     ivaca=20;

300     if (ivaca>ivacap) /* check peak value */
301     {
302     clrthres(ivacap,0);
303     ivacap=ivaca;
304     ivacptc=ivacptc;
305     setthres(ivacap,0);
306     }
307     if (ivaca==ivacla)
308     goto accapdsp;
309     if (ivaca>ivacla)
310     {
311     for (i=ivacla+1;i<=ivaca;i++)
312     setvacc(i,0);
313     }
314     if (ivaca<ivacla)
315     {
316     for (i=ivacla;i>ivaca;i--)
317     clrvacc(i,0);
318     }
319     ivacla=ivaca;

320     accapdsp:    if (ivacap<=0)      /* peak AFT acceleration display */
321     goto accdsp;
322     ivacptc--;
323     if (ivacptc==0)
324     {
325     ivacptc=ivacptc;
326     clrthres(ivacap,0);
327     ivacap--;
328     setthres(ivacap,0);
329     }
330     goto accdsp;

331     /* accelerometer gain */

332     accgain:    if(iaccgn==1)
333     iaccgn=2;

```

```

334     else
335     iaccgn=1;
336     goto accdspd;

337     /* system function setup */

338     sysfunct:  clear();
339     menu();
340     waitpb();
341     if (irecord==0)
342     lcdtext(204,108,sstart);
343     else
344     lcdtext(206,108,sstop);
345     lcdtext(201,96,srecord);
346     lcdtext(204,76,sstart);
347     lcdtext(206,64,sdata);
348     lcdtext(204,38,ssetup);
349     lcdtext(206,6,sexit);
350     strcpy(string,">Select RECORD, DATA,");
351     lcdtext(0,12,string);
352     strcpy(string," SETUP or EXIT");
353     lcdtext(0,0,string);

354     /* system function display */

355     sysdsp:    readpb(&ipb);      /* 0 none
356     start recording
357     start data transfer
358     setup (diagnostics)
359     exit */
360     if (ipb==1)
361     goto startrec;
362     if (ipb==2)
363     goto startdata;
364     if (ipb==3)
365     goto setdiag;
366     if (ipb==4)
367     goto accdspd;
368     goto sysdsp;

369     /* heading/azimuth display setup */

370     hdazmdspd: clear();
371     menu();
372     waitpb();
373     lcdtext(204,102,saccel);
374     lcdtext(201,6,ssystem);
375     distxt(142,104,48);
376     distxt(189,61,57);
377     distxt(189,50,48);
378     lcdtext(136,6,s180);
379     distxt(95,66,50);
380     distxt(95,55,55);
381     distxt(95,44,48);

```

```

382     disphdazm();
383     icmpsl=-1;          /* last compass */
384     iptchl=0;          /* last pitch */
385     irotl=0;          /* last rotate */
386     ihdazmd=1;        /* hd/azm display toggle */

387     /* heading/azimuth display */

388     hdazmdsp:   readpb(&ipb);      /* 0 none
389     accel display
390     system */
391     hdazmpb:    if (ipb==1)
392     goto accdspd;
393     if (ipb==4)
394     goto sysfunt;

395     wclock();        /* sample hd/azm data */
396     icmps=0;
397     irot=0;
398     iptch=0;
399     for (j=1;j<=6;j++)
400     {
401         adc2(4,&iacc[11]);
402         icmps=icmps+iacc[11];
403     }
404     icmps=icmps/6;
405     for (j=1;j<=10;j++)
406     {
407         adc2(5,&iacc[11]);
408         adc2(6,&iacc[12]);
409         iptch=iptch-iacc[11]+iptchz; /* reverse for display */
410         irot=irot-iacc[12]+irotz;
411     }
412     iptch=iptch/10;
413     irot=irot/10;

414     if (icmpsl===-1)
415     icmps=icmps;
416     readpb(&ipb);
417     if (ipb!=0)
418     goto hdazmpb;
419     showhd(icmps,icmpsl);
420     icmps=icmps;
421     readpb(&ipb);
422     if (ipb!=0)
423     goto hdazmpb;
424     showazm(iptch,iptchl,irot,irotl);
425     iptchl=iptch;
426     irotl=irot;
427     goto hdazmdsp;

428     /* start recording */

429     startrec:   clear();

```

```

430     menu();
431     waitpb();
432     strcpy(string,">Select UNCOMPpressed or");
433     lcdtext(0,12,string);
434     strcpy(string," COMPressed recording mode");
435     lcdtext(0,0,string);
436     irecord=0;
437     lcdtext(201,108,srecord);
438     lcdtext(201,96,suncomp);
439     lcdtext(201,76,srecord);
440     lcdtext(206,64,scomp);
441     lcdtext(206,6,sexit);

442     strtrec:   readpb(&ipb);      /* 0 none
443     uncompressed
444     compressed
445     exit */
446     if (ipb==1)
447     goto uncomprec;
448     if (ipb==2)
449     goto comprec;
450     if (ipb==4)
451     goto accdsps;
452     goto strtrec;

453     uncomprec:   icomp=0;
454     goto uncmprec;
455     comprec:     icomp=1;
456     uncmprec:    lrecnum=0;
457     irecord=1;
458     fbuffer[0]=icomp;
459     j=2;
460     for (i=1;i<=9;i++)
461     {
462     fbuffer[j]=ichz[i];
463     j++;
464     j++;
465     fbuffer[j]=ichg[i];
466     j++;
467     j++;
468     }
469     fbuffer[j]=iptchz;
470     j++;
471     j++;
472     fbuffer[j]=irotz;
473     if(write(ifhandle, fbuffer, 24)!=24) /* data recording */
474     +goto recerr;
475     goto accdsps;

476     /* stop recording */

477     stoprec:     irecord=0;
478     close(ifhandle);
479     clear();

```

```

480     strcpy(string,">RECORDING stopped");
481     lcdtext(0,12,string);
482     strcpy(string," file: ");
483     lcdtext(0,0,string);
484     lcdtext(40,0,sfile);
485     wait(5000);
486     goto restart;

487     /* start data transfer */

488     startdata:   clear();
489     menu();
490     waitpb();
491     lcdtext(204,108,sstart);
492     lcdtext(206,96,sdata);
493     lcdtext(206,6,sexit);
494     strcpy(string,">Ready for DATA transfer");
495     lcdtext(0,12,string);
496     strcpy(string," START or EXIT");
497     lcdtext(0,0,string);
498     lcdtext(206,6,sexit);

499     strtdat:     readpb(&ipb);      /* 0 none
500     exit */
501     if (ipb==1)
502     goto strtdatc;
503     if (ipb==4)
504     goto accdsps;
505     goto strtdat;

506     strtdatc:    clear();
507     waitpb();
508     strcpy(string,">DATA transfer mode,");
509     lcdtext(0,12,string);
510     strcpy(string,"begin remote session");
511     lcdtext(0,0,string);
512     goto extpgm;

513     /* setup (diagnostics) */

514     setdiag:     clear();
515     menu();
516     waitpb();
517     lcdtext(204,102,saccel);
518     lcdtext(201,70,sdisply);
519     lcdtext(206,6,sexit);
520     strcpy(string,">Select ACCElrometer or");
521     lcdtext(0,12,string);
522     strcpy(string," DISPLAY setup");
523     lcdtext(0,0,string);

524     stdia:       readpb(&ipb);      /* 0 none
525     accelerometers
526     display

```

```

527      exit */
528      if (ipb==1)
529          goto setaccel;
530      if (ipb==2)
531          goto setdisply;
532      if (ipb==4)
533          goto accdsps;
534      goto stdia;

535      /* setup accelerometer */

536      setaccel:    clear();
537      menu();
538      waitpb();
539      lcdtext(206,6,sexit);
540      strcpy(string,"Peak Triaxial Acceleration");
541      lcdtext(0,110,string);
542      strcpy(string,"HELMSMAN");
543      lcdtext(0,80,string);
544      strcpy(string,"STARBOARD AFT");
545      lcdtext(0,50,string);
546      strcpy(string,"PORT AFT");
547      lcdtext(0,20,string);

548      staccl:      readpb(&ipb);      /* 4 exit */

549      if (ipb==4)
550          goto accdsps;

551      for (i=1;i<=6;i++)
552          adc1(i,&iacc[i]);
553      for (i=1;i<=3;i++)
554          adc2(i,&iacc[i+6]);

555      for (i=65;i<=78;i++)
556          clrline(150,i);

557      readpb(&ipb);
558      if (ipb==4)
559          goto accdsps;

560      ftemp=(iacc[1]-ichz[1])/fchg[1];
561      sprintf(string,"X %5.1f",ftemp);
562      lcdtext(0,65,string);
563      ftemp=(iacc[2]-ichz[2])/fchg[2];
564      sprintf(string,"Y %5.1f",ftemp);
565      lcdtext(50,65,string);
566      ftemp=(iacc[3]-ichz[3])/fchg[3];
567      sprintf(string,"Z %5.1f [G]",ftemp);
568      lcdtext(100,65,string);

569      readpb(&ipb);
570      if (ipb==4)
571          goto accdsps;

```

```

572     for (i=35;i<=48;i++)
573         clrline(150,i);

574     readpb(&ipb);
575     if (ipb==4)
576         goto accdsps;

577     ftemp=(iacc[4]-ichz[4])/fchg[4];
578     sprintf(string,"X %5.1f",ftemp);
579     lcdtext(0,35,string);
580     ftemp=(iacc[5]-ichz[5])/fchg[5];
581     sprintf(string,"Y %5.1f",ftemp);
582     lcdtext(50,35,string);
583     ftemp=(iacc[6]-ichz[6])/fchg[6];
584     sprintf(string,"Z %5.1f [G]",ftemp);
585     lcdtext(100,35,string);

586     readpb(&ipb);
587     if (ipb==4)
588         goto accdsps;

589     for (i=5;i<=18;i++)
590         clrline(150,i);

591     readpb(&ipb);
592     if (ipb==4)
593         goto accdsps;

594     ftemp=(iacc[7]-ichz[7])/fchg[7];
595     sprintf(string,"X %5.1f",ftemp);
596     lcdtext(0,5,string);
597     ftemp=(iacc[8]-ichz[8])/fchg[8];
598     sprintf(string,"Y %5.1f",ftemp);
599     lcdtext(50,5,string);
600     ftemp=(iacc[9]-ichz[9])/fchg[9];
601     sprintf(string,"Z %5.1f [G]",ftemp);
602     lcdtext(100,5,string);

603     goto stacc1;

604     /* setup display */

605     setdisply:   clear();
606     menu();
607     waitpb();
608     lcdtext(201,102,sadjust);
609     lcdtext(204,72,sclear);
610     lcdtext(206,6,sexit);
611     strcpy(string,">Select ADJUST or");
612     lcdtext(0,12,string);
613     strcpy(string," CLEAR display");
614     lcdtext(0,0,string);

```

```

615      stdisp:      readpb(&ipb);      /* 0 none
616      adjust
617      clear display
618      exit */
619      if (ipb==1)
620      goto stdadj;
621      if (ipb==2)
622      goto stdclr;
623      if (ipb==4)
624      goto accdsps;
625      goto stdisp;

626      stdadj:      clear();
627      menu();
628      waitpb();
629      lcdtext(201,102,scntrst);
630      lcdtext(206,6,sexit);
631      strcpy(string,">ADJUST display contrast");
632      lcdtext(0,0,string);

633      sdadj:      readpb(&ipb);      /* 0 none
634      contrast
635      exit */
636      if (ipb==1)
637      {
638      lcdinc();
639      ilcd++;
640      if (ilcd==43)
641      {
642      for (i=44;i<=63;i++)
643      lcdinc();
644      for (i=0;i<=25;i++)
645      lcdinc();
646      ilcd=25;
647      }
648      wait(125);
649      }

650      if (ipb==4)
651      goto accdsps;

652      goto sdadj;

653      /* setup display clear */

654      stdclr:      clear();
655      waitpb();
656      strcpy(string,">Wait for display to CLEAR");
657      lcdtext(0,0,string);
658      wait(1000);
659      for (i=ilcd;i<=63;i++)
660      lcdinc();
661      lcdinc();
662      lcdoff();

```

```

663     wait(8000);
664     lcdmid();
665     for (i=33;i<=63;i++)
666         lcdinc();
667     lcdinc();
668     for (i=1;i<=ilcd;i++)
669         lcdinc();
670     goto setdisply;

671     /* error */

672     erraccel:    strcpy(string,">ERROR in calibration file");
673     goto errexit;
674     errfopen:    strcpy(string,">ERROR in opening data file");
675     goto errexit;
676     recerr:      strcpy(string,">ERROR in recording data");

677     errexit:    clear();
678     lcdtext(0,12,string);
679     strcpy(string," run remote session for diagnosis");
680     lcdtext(0,0,string);
681     wait(20000);
682     extpgm:     close(ifhandle);
683     printf("\nHRA v1.2 exit\007");
684     }
685     /* end of main */

686     /* subroutines */

687     /* wait msec */
688     wait(int idsec)
689     {
690         int istp;
691         istp=idsec/20;    /* 50 Hz RTC */
692         for(i=1;i<=istp;i++)
693             wclock();
694     }

695     /* display string on LCD starting at ix, iy */
696     lcdtext(int ix, int iy, char *str)
697     {
698         int i,ic,il;
699         char khar;

700         il=strlen(str);
701         for (i=0;i<=il;i++)
702         {
703             khar=str[i];
704             ic=khar;
705             if (ic==0)
706                 break;
707             distxt(ix,iy,ic);
708             ix=ix+6;
709         }

```

```

710     }

711     /* rectangle display at lower left location ix,iy with width iw and
712     height ih */
713     rectangle(int ixl, int iyl, int iw, int ih)
714     {
715         move(ixl,iyl);
716         draw(ixl+iw,iyl);
717         draw(ixl+iw,iyl+ih);
718         draw(ixl,iyl+ih);
719         draw(ixl,iyl);
720     }

721     /* button menu */
722     menu()
723     {
724         move(198,0);
725         draw(198,127);
726         rectangle(199,0,40,31);
727         rectangle(199,32,40,31);
728         rectangle(199,64,40,31);
729         rectangle(199,96,40,31);
730     }

731     /* circle display at center ixc,iyc with radius ir*/
732     circle(int ixc,int iyc,int ir)
733     {
734         int ix,iy,iangle;
735         double fx,fy,fxc,fyc,fangle,fr,radfactor;

736         radfactor=6.283185307/360.;
737         fxc=ixc;
738         fyc=iyc;
739         fr=ir;
740         fx=fxc+fr;
741         ix=fx;
742         iy=fyc;
743         move(ix,iy);
744         for (iangle=0;iangle<=360;iangle++)
745         {
746             fangle=iangle;
747             fangle=radfactor*fangle;
748             fx=fxc+fr*cos(fangle)+0.5;
749             fy=fyc+fr*sin(fangle)+0.5;
750             ix=fx;
751             iy=fy;
752             draw(ix,iy);
753         }
754     }

755     /* heading/azimuth display */
756     disphdazm()
757     {
758         char buffer[6];

```

```

759      int i,ix,iy;

760      circle(50,64,40);
761      circle(50,64,41);
762      move(10,64);
763      draw(90,64);
764      move(10,63);
765      draw(90,63);
766      move(48,62);
767      draw(52,62);
768      move(49,61);
769      draw(51,61);
770      setpix(50,60);
771      circle(145,64,40);
772      circle(145,64,41);
773      for (i=0;i<=2;i++)
774      {
775          move(144+i,104);
776          draw(144+i,99);
777          move(144+i,24);
778          draw(144+i,29);
779          move(105,63+i);
780          draw(110,63+i);
781          move(180,63+i);
782          draw(185,63+i);
783      }
784      }

785      /* show heading */
786      showhd(int icmps,int icmpsl)
787      {
788          int ix,iy;
789          double fhd,fx,fy,fr,radfactor;

790          radfactor=6.283185307/360.;
791          fr=35.;
792          fhd=icmpsl/11.37;
793          fhd=radfactor*fhd;
794          fy=fr*cos(fhd)+0.5;
795          fx=fr*sin(fhd)+0.5;
796          ix=fx;
797          ix=ix+145;
798          iy=fy;
799          iy=iy+64;
800          move(145,64);
801          erase(ix,iy);
802          fhd=icmps/11.37;
803          fhd=radfactor*fhd;
804          fy=fr*cos(fhd)+0.5;
805          fx=fr*sin(fhd)+0.5;
806          ix=fx;
807          ix=ix+145;
808          iy=fy;
809          iy=iy+64;

```

```

810     move(145,64);
811     draw(ix,iy);
812 }

813     /* show azimuth */
814     showazm(int iptch, int iptchl, int irot, int irotl)
815     {
816         int ixl,ixll,iyl,iyll,ixr,ixrl,iyr,iyrl,iyd,iydl;
817         double fptch,frot,fb,fc,fx,fy,fyd,fr,fradius,fpi,fpi2,radfactor;
818         double fsrot,fcrot,fspirota,fspirotb;

819         radfactor=6.283185307/360.;
820         fpi=3.141592654;
821         fpi2=1.570796327;
822         fradius=35.;

823         fptch=iptchl;
824         fyd=fptch/34.13;
825         iyd=fyd;
826         iyd=64-iyd;
827         frot=irotl;
828         frot=radfactor*(frot/34.13);
829         fspirota=sin(fpi2-frot);
830         fspirotb=sin(fpi2+frot);
831         fcrot=cos(frot);
832         fsrot=sin(frot);

833         fb=asin(fyd*fspirota/fradius);
834         fc=fpi2+frot-fb;
835         fr=fradius*sin(fc)/fspirota;
836         fx=fr*fcrot+0.5;
837         fy=fr*fsrot+0.5;
838         ixrl=fx;
839         iyrl=fy;
840         iyrl=iyrl+iyd;

841         fb=asin(fyd*fspirotb/fradius);
842         fc=fpi/2-frot-fb;
843         fr=fradius*sin(fc)/fspirotb;
844         fx=fr*fcrot+0.5;
845         fy=fr*fsrot+0.5;
846         ixl=fx;
847         iyll=fy;
848         iyll=iydl-iyll;

849         move(10,64);
850         draw(90,64);
851         move(10,63);
852         draw(90,63);
853         move(48,62);
854         draw(52,62);
855         move(49,61);
856         draw(51,61);

```

```

857      fptch=iptch;
858      fyd=fptch/34.13;
859      iyd=fyd;
860      iyd=64-iyd;
861      frot=irot;
862      frot=radfactor*(frot/34.13);

863      fspirot=sin(fpi2-frot);
864      fspirotb=sin(fpi2+frot);
865      fcrot=cos(frot);
866      fsrot=sin(frot);

867      fb=asin(fyd*fspirot/radius);
868      fc=fpi2+frot-fb;
869      fr=radius*sin(fc)/fspirot;
870      fx=fr*fcrot+0.5;
871      fy=fr*fsrot+0.5;
872      ixr=fx;
873      iyr=fy;
874      iyr=iyr+iyd;

875      fb=asin(fyd*fspirotb/radius);
876      fc=fpi2-frot-fb;
877      fr=radius*sin(fc)/fspirotb;
878      fx=fr*fcrot+0.5;
879      fy=fr*fsrot+0.5;
880      ixl=fx;
881      iyl=fy;
882      iyl=iyd-iyl;

883      if (ixr!=ixr)
884      goto shazmc;
885      if (iyr!=iyr)
886      goto shazmc;
887      if (ixl!=ixl)
888      goto shazmc;
889      if (iyl!=iyl)
890      goto shazmc;
891      if (iyd==iyd)
892      goto nshazm;

893      shazmc:      clrpix(50,iyd);
894      move(50,iyd);
895      erase(50+ixr,iyr);
896      move(50,iyd);
897      erase(50-ixl,iyl);
898      clrpix(50,iyd);
899      move(50,iyd);
900      draw(50+ixr,iyr);
901      move(50,iyd);
902      draw(50-ixl,iyl);

903      nshazm:      move (50,50);
904      }

```

```

905      /* acceleration display */
906      dispacc(int igain)
907      {
908      char buffer[6];
909      int i,ixx,ix,iy;

910      for (i=0;i<=20;i++)
911      {
912      iy=i*6+2;
913      rectangle(55,iy,20,4);
914      rectangle(115,iy,20,4);
915      }
916      for (iy=3;iy<=5;iy++)
917      {
918      move(55,iy);
919      draw(75,iy);
920      move(115,iy);
921      draw(135,iy);
922      }
923      setthres(0,0);
924      setthres(0,1);
925      strcpy(buffer,"AFT");
926      lcdtext(20,110,buffer);
927      strcpy(buffer,"HELM");
928      lcdtext(153,110,buffer);
929      if (igain==2)
930      goto dispaccg;

931      for (i=0;i<=8;i=i+2)
932      {
933      iy=i*12-4;
934      itoa(i,buffer,10);
935      lcdtext(92,iy,buffer);
936      move(85,iy+8);
937      draw(90,iy+8);
938      move(100,iy+8);
939      draw(106,iy+8);
940      }
941      move(85,124);
942      draw(88,124);
943      move(103,124);
944      draw(106,124);
945      strcpy(buffer,"10");
946      lcdtext(89,113,buffer);
947      goto dispaccr;

948      dispaccg:    for (i=0;i<=4;i++)
949      {
950      iy=i*24-4;
951      itoa(i,buffer,10);
952      lcdtext(92,iy,buffer);
953      move(85,iy+8);
954      draw(90,iy+8);

```

```

955     move(100,iy+8);
956     draw(106,iy+8);
957     }
958     move(85,124);
959     draw(90,124);
960     move(100,124);
961     draw(106,124);
962     itoa(5,buffer,10);
963     lcdtext(92,113,buffer);
964     dispaccr:   move(100,100);
965     }

966     /* set vertical acceleration display box at position ivac (1-20)
967     and for left (0) or right (1) display */
968     setvacc(int ivac,int idisp)
969     {
970     int i,iy,ixx,iyy;

971     ixx=56+idisp*60;
972     iyy=ivac*6+2;
973     for (i=1;i<=3;i++)
974     {
975     iy=iyy+i;
976     move(ixx,iy);
977     draw(ixx+18,iy);
978     }
979     }

980     /* clear vertical acceleration display box at position ivac (1-20)
981     and for left (0) or right (1) display */
982     clrvacc(int ivac, int idisp)
983     {
984     int i,iy,ixx,iyy;

985     ixx=56+idisp*60;
986     iyy=ivac*6+2;
987     for (i=1;i<=3;i++)
988     {
989     iy=iyy+i;
990     move(ixx,iy);
991     erase(ixx+18,iy);
992     }
993     }

994     /* set threshold triangular mark at position ivac (1-20)
995     and for left (0) or right (1) display */

996     setthres(int ivac, int idisp)
997     {
998     int i,ixx,iyy;

999     ixx=49+idisp*60;
1000    iyy=ivac*6+4;
1001    move(ixx,iyy-3);

```

```

1002     draw(ixx,iyy+3);
1003     ixx++;
1004     move(ixx,iyy-2);
1005     draw(ixx,iyy+2);
1006     ixx++;
1007     move(ixx,iyy-1);
1008     draw(ixx,iyy+1);
1009     ixx++;
1010     setpix(ixx,iyy);
1011     ixx=ixx+26;
1012     setpix(ixx,iyy);
1013     ixx++;
1014     move(ixx,iyy-1);
1015     draw(ixx,iyy+1);
1016     ixx++;
1017     move(ixx,iyy-2);
1018     draw(ixx,iyy+2);
1019     ixx++;
1020     move(ixx,iyy-3);
1021     draw(ixx,iyy+3);
1022     }

1023     /* clear threshold triangular mark at position ivac (1-20)
1024     and for left (0) or right (1) display */

1025     clrthres(int ivac, int idisp)
1026     {
1027     int i,ixx,iyy;

1028     ixx=49+idisp*60;
1029     iyy=ivac*6+4;
1030     move(ixx,iyy-3);
1031     erase(ixx,iyy+3);
1032     ixx++;
1033     move(ixx,iyy-2);
1034     erase(ixx,iyy+2);
1035     ixx++;
1036     move(ixx,iyy-1);
1037     erase(ixx,iyy+1);
1038     ixx++;
1039     clrpx(ixx,iyy);
1040     ixx=ixx+26;
1041     clrpx(ixx,iyy);
1042     ixx++;
1043     move(ixx,iyy-1);
1044     erase(ixx,iyy+1);
1045     ixx++;
1046     move(ixx,iyy-2);
1047     erase(ixx,iyy+2);
1048     ixx++;
1049     move(ixx,iyy-3);
1050     erase(ixx,iyy+3);
1051     }
1052     /* end of helm.c */

```

```

1      ;Helmsman Recording Accelerometer subroutines
2      ;helmsub.asm
3      ;
4      ;c 1998 Conrad Technologies, Paoli PA
5      ;
6      PIAPTA equ 50H ;PIA port A
7      PIAPTB equ 51H ;PIA port B
8      PIAPTC equ 52H ;PIA port C
9      PIACTR equ 53H ;PIA control register
10     ;
11     ADC1SR equ 110H ;ADC1 status register
12     ADC1CC equ 111H ;ADC1 conversion command
13     ADC1LS equ 111H ;ADC1 LSB data
14     ADC1MS equ 112H ;ADC1 MSB data
15     ;
16     ADC2SR equ 114H ;ADC2 starts register
17     ADC2CC equ 115H ;ADC2 conversion command
18     ADC2LS equ 115H ;ADC2 LSB data
19     ADC2MS equ 116H ;ADC2 MSB data
20     ;
21     LCR equ 3FBH ;line control register
22     BRDL equ 3F8H ;baud rate divisor low
23     BRDH equ 3F9H ;baud rate divisor high
24     MCR equ 3FCH ;modem control register
25     LSR equ 3FDH ;line status register
26     RDR equ 3F8H ;receive data register
27     ;
28     ;LCD
29     ;inverted B0 WR write output
30     ;command B1 RD read output
31     ;signals B2 CE chip enable output
32     ; B3 /C D command / data output
33     ; B4 RESET controller reset output
34     ; B5 ADJ contrast adjustment output
35     ; B6 CTRL contrast control output
36     ; B7 RTCE real-time clock echo output
37     ;
38     ; A0-A7 D0-D7 bidirectional data input/output
39     ;
40     ;Pushbuttons
41     ; C0-C3 PB1-PB4 inputs
42     ;
43     ;Real Time Clock
44     ; C4 CLOCK input
45     ;
46     .model small
47     .code
48     ;
49     ;read ADC1 channel adc1(ichan,&idata)
50     ; ichan 1-8
51     ;
52     public _adc1
53     _adc1 proc
54     push bp ;save bp register

```

```

55      mov bp,sp
56      mov ax,ADC1CC
57      mov dx,ax
58      mov ax,[bp+4]      ;ichan
59      dec ax              ;ichan=ichan-1
60      out dx,al
61      mov ax,ADC1SR
62      mov dx,ax
63      adc1wt: in al,dx
64      and al,3
65      cmp al,3
66      jne adc1wt
67      mov ax,ADC1LS
68      mov dx,ax
69      in al,dx
70      mov bl,al
71      mov ax,ADC1MS
72      mov dx,ax
73      in al,dx
74      mov ah,al
75      mov al,bl
76      mov cl,4
77      ror ax,cl
78      and ax,0FFFH
79      mov bx,[bp+6]      ;idata offset
80      mov [bx],ax
81      pop bp
82      ret
83      _adc1      endp
84      ;
85      ;read ADC2 channel      adc2(ichan,&idata)
86      ;
87      public _adc2
88      _adc2      proc
89      push bp          ;save bp register
90      mov bp,sp
91      mov ax,ADC2CC
92      mov dx,ax
93      mov ax,[bp+4]      ;ichan
94      dec ax              ;ichan=ichan-1
95      out dx,al
96      mov ax,ADC2SR
97      mov dx,ax
98      adc2wt: in al,dx
99      and al,3
100     cmp al,3
101     jne adc2wt
102     mov ax,ADC2LS
103     mov dx,ax
104     in al,dx
105     mov bl,al
106     mov ax,ADC2MS
107     mov dx,ax
108     in al,dx

```

```

109      mov ah,al
110      mov al,bl
111      mov cl,4
112      ror ax,cl
113      and ax,0FFFH
114      mov bx,[bp+6]      ;idata offset
115      mov [bx],ax
116      pop bp
117      ret
118      _adc2      endp
119      ;
120      ;read pushbutton      readpb(&idata)
121      ;
122      public _readpb
123      _readpb      proc
124      push bp      ;save bp register
125      mov bp,sp
126      mov ax,PIAPTC
127      mov dx,ax
128      in al,dx
129      and al,0FH
130      cmp al,8
131      je rdpb4
132      cmp al,4
133      je rdpb3
134      cmp al,2
135      je rdpb2
136      cmp al,1
137      je rdpb1
138      mov al,0
139      jmp rdpbx

140      rdpb4:      mov al,4
141      jmp rdpbx
142      rdpb3:      mov al,3
143      jmp rdpbx
144      rdpb2:      mov al,2
145      jmp rdpbx
146      rdpb1:      mov al,1

147      rdpbx:      mov ah,0
148      mov bx,[bp+4]      ;idata offset
149      mov [bx],ax
150      pop bp
151      ret
152      _readpb      endp
153      ;
154      ;wait for pushbutton release      waitpb();
155      ;
156      public _waitpb
157      _waitpb      proc
158      mov ax,PIAPTC
159      mov dx,ax
160      wtpb1:      in al,dx

```

```

161      and al,0FH
162      jnz wtpbl
163      ret
164      _waitpb endp
165      ;
166      ;read real-time clock          clock(&iclock)
167      ;
168      public _clock
169      _clock proc
170      push bp          ;save bp register
171      mov bp,sp
172      mov ax,PIAPTC
173      mov dx,ax
174      in al,dx
175      and al,10H
176      mov bx,[bp+4]
177      mov [bx],ax
178      pop bp
179      ret
180      _clock endp
181      ;
182      ;wait for real time clock      wclock();
183      ;
184      public _wclock
185      _wclock proc
186      mov ax,PIAPTC
187      mov dx,ax
188      wclkd: in al,dx          ;wait if high
189      and al,10H
190      jnz wclkd
191      wclk:  in al,dx          ;wait for high
192      and al,10H
193      jz wclk
194      mov ax,PIAPTB
195      mov dx,ax
196      mov al,cs:b765dat      ;toggle b7 RTC echo
197      or al,10000000b
198      out dx,al
199      and al,7FH
200      out dx,al
201      ret
202      _wclock endp
203      ;
204      ;initialize hardware          inithwd();
205      ;
206      public _inithwd
207      _inithwd proc
208      mov ax,2              ;init LCD twice
209      mov di,ax
210      mov al,01000000b      ;shutdown LCD bias
211      mov cs:b765dat,al
212      intdspl: mov ax,PIACTR      ;initialize LCD
213      mov dx,ax
214      mov al,10001001b      ;active, mode 0, PTA out,

```

```

215      out dx,al          ; PTB out, PTC in
216      mov ax,PIAPTb
217      mov dx,ax
218      mov al,01000000b   ;B4 reset release + b765dat
219      out dx,al
220      mov al,01010000b   ;B4 reset + b765dat
221      out dx,al
222      mov cx,0FFFFh      ;1 msec delay
223      intdslp: loop intdslp
224      mov al,01000000b   ;B4 reset release + b765dat
225      out dx,al
226      mov bl,10000000b   ;mode set
227      call discmd
228      mov bl,0           ;text home 1000h
229      call disdat
230      mov bl,00010000b
231      call disdat
232      mov bl,01000000b
233      call discmd
234      mov bl,00101000b   ;text area 0028h
235      call disdat
236      mov bl,00000000b
237      call disdat
238      mov bl,01000001b
239      call discmd
240      mov bl,0           ;graphic home 0000h
241      call disdat
242      mov bl,0
243      call disdat
244      mov bl,01000010b
245      call discmd
246      mov bl,00101000b   ;graphic area 0028h
247      call disdat
248      mov bl,0
249      call disdat
250      mov bl,01000011b
251      call discmd
252      mov bl,1001000b    ;display OFF
253      call discmd
254      dec di
255      jnz intdspl
256      ret
257      _inithwd endp
258      ;
259      ;set LCD bias to midscale      lcdmid()
260      ;
261      public _lcdmid
262      _lcdmid proc
263      mov ax,PIAPTb
264      mov dx,ax
265      mov al,cs:b765dat
266      or al,01100000b     ;midscale 11
267      out dx,al
268      mov al,cs:b765dat

```

```

269      and al,80H          ;normal 00
270      out dx,al
271      mov cs:b765dat,al
272      ret
273      _lcdmid endp
274      ;
275      ;increment LCD bias      lcdinc()
276      ;
277      public _lcdinc
278      _lcdinc proc
279      mov ax,PIAPTB
280      mov dx,ax
281      mov al,cs:b765dat
282      and al,80H          ;increment 00 01 00
283      out dx,al
284      or al,00100000b
285      out dx,al
286      and al,80H          ;normal 00
287      out dx,al
288      mov cs:b765dat,al
289      ret
290      _lcdinc endp
291      ;
292      ;shutdown LCD bias      lcdoff()
293      ;
294      public _lcdoff
295      _lcdoff proc
296      mov ax,PIAPTB
297      mov dx,ax
298      mov al,cs:b765dat
299      or al,01000000b      ;shutdown 10
300      out dx,al
301      mov al,cs:b765dat
302      ret
303      _lcdoff endp
304      ;
305      ;clear graphic display    clear()
306      ;
307      public _clear
308      _clear proc
309      mov bl,0
310      call disdat
311      mov bl,0
312      call disdat
313      mov bl,00100100b      ;address pointer set
314      call discmd
315      mov cx,1BFFh          ;clear text, attribute, graphics
316      clear: mov bl,0
317      call disdat
318      mov bl,11000000b      ;write data
319      call discmd
320      loop clear
321      mov cx,5120            ;display 40*128 = 5120 bytes
322      mov bx,offset image    ;display image

```

```

323     mov al,0
324     cleari: mov cs:[bx],al
325     inc bx
326     loop cleari
327     ret
328     _clear     endp
329     ;
330     ;display ON          dispon()
331     ;
332     public _dispon
333     _dispon proc
334     mov bl,10011000b      ;display ON
335     call discmd
336     ret
337     _dispon     endp
338     ;
339     ;display OFF        dispoff()
340     ;
341     public _dispoff
342     _dispoff proc
343     mov bl,10010000b      ;display ON
344     call discmd
345     ret
346     _dispoff     endp
347     ;
348     ;set pixel          setpix(ix,iy)
349     ;
350     public _setpix
351     _setpix proc
352     push bp                ;save bp register
353     mov bp,sp
354     call setclrp
355     or al,ah
356     mov cs:[bx],al        ;write image data
357     ;
358     mov bl,al              ;display
359     call disdat
360     mov bl,11000000b      ;write data
361     call discmd
362     pop bp
363     ret
364     _setpix     endp
365     ;
366     ;clear pixel        clrpix(ix,iy)
367     ;
368     public _clrpix
369     _clrpix proc
370     push bp                ;save bp register
371     mov bp,sp
372     call setclrp
373     xor al,ah
374     mov cs:[bx],al        ;write image data
375     ;
376     mov bl,al              ;display

```

```

377      call disdat
378      mov bl,11000000b      ;write data
379      call discmd
380      pop bp
381      ret
382      _clrpix endp
383      ;
384      ;clear line      clrline(in,iy)
385      ;in number of pixels to clear, iy line number, assume ix=0 (start of line)
386      ;
387      public _clrline
388      _clrline proc
389      push bp      ;save bp register
390      mov bp,sp
391      mov ax,[bp+4]      ;in
392      mov cx,ax
393      mov ax,0      ;ix=0
394      mov dx,ax
395      mov ax,[bp+6]      ;iy
396      ;
397      clrlnl: push cx      ;in
398      push ax      ;iy
399      push dx      ;ix
400      call stclrp
401      not ah
402      and al,ah
403      mov cs:[bx],al      ;write image data
404      ;
405      mov bl,al      ;display
406      call disdat
407      mov bl,11000000b      ;write data
408      call discmd
409      pop dx      ;ix
410      pop ax      ;iy
411      pop cx      ;in
412      dec cx
413      jz clrlnx
414      inc dx
415      jmp clrlnl
416      clrlnx: pop bp
417      ret
418      _clrline endp

419      ;
420      ;move to ix,iy      move(ix,iy)
421      ;
422      public _move
423      _move proc
424      push bp      ;save bp register
425      mov bp,sp
426      mov ax,[bp+4]      ;ix
427      mov cs:xpos,ax
428      mov ax,[bp+6]      ;iy
429      mov cs:ypos,ax

```

```

430     pop bp
431     ret
432     _move     endp
433     ;
434     ;draw to ix,iy      draw(ix,iy)
435     ;
436     public     _draw
437     _draw     proc
438     push bp          ;save bp register
439     mov bp,sp
440     mov ax,[bp+4]      ;ix
441     mov cs:x2,ax
442     mov ax,[bp+6]      ;iy
443     mov cs:y2,ax
444     mov ax,1
445     mov cs:drweraf,ax
446     call drwera
447     pop bp
448     ret
449     _draw     endp
450     ;
451     ;erase to ix,iy      erase(ix,iy)
452     ;
453     public     _erase
454     _erase     proc
455     push bp          ;save bp register
456     mov bp,sp
457     mov ax,[bp+4]      ;ix
458     mov cs:x2,ax
459     mov ax,[bp+6]      ;iy
460     mov cs:y2,ax
461     mov ax,0
462     mov cs:drweraf,ax
463     call drwera
464     pop bp
465     ret
466     _erase     endp
467     ;
468     ;display text data at position ix,iy  distxt(ix,iy,ic)
469     ;
470     public     _distxt
471     _distxt     proc
472     push bp          ;save bp register
473     mov bp,sp
474     mov ax,[bp+4]      ;ix
475     mov cs:xtpos,ax
476     mov ax,[bp+6]      ;iy
477     mov cs:ytpos,ax
478     mov ax,0000Eh      ;shape table count = 14
479     mov cs:stccnt,ax
480     mov dx,ax
481     mov ax,[bp+8]      ;ic
482     mul di

```

```

483      add ax,14          ;reverse char shape
484      mov di,ax          ;char*shape table count
485      dstxst: mov cx,7    ;pixel count
486      mov ah,cs:chrdata[di]
487      ;
488      dstnxp: rol ah,1
489      jnc dstxskp
490      ;
491      push ax
492      mov ax,cs:xtpos
493      mov dx,ax
494      mov ax,cs:ytpos
495      call stclrp
496      or al,ah
497      mov cs:[bx],al      ;write image data
498      mov bl,al           ;display
499      call disdat
500      mov bl,11000000b    ;write data
501      call discmd
502      pop ax
503      ;
504      dstxskp: inc cs:xtpos ;next pixel
505      loop dstnxp          ;loop until 8 pixels
506      inc cs:ytpos        ;next scan line
507      dec di              ;next character code
508      sub cs:xtpos,7       ;reset x position
509      mov ax,cs:stccnt
510      dec ax               ;decrement scan line count
511      mov cs:stccnt,ax
512      jne dstxst          ;until finished
513      pop bp              ;restore bp
514      ret
515      _distxt endp
516      ;
517      ;convert ix,iy to display address, set address pointer, read
518      ;image data as al, set ah to bit position, bx as pointer
519      ;to image data
520      ;
521      ;ix, iy on stack
522      ;
523      setclrp proc
524      mov ax,[bp+4]        ;ix
525      mov dx,ax
526      mov ax,[bp+6]        ;iy
527      call stclrp
528      ret
529      setclrp endp
530      ;
531      ;dx = ix, ax = iy
532      ;
533      stclrp proc
534      mov bl,40
535      mul bl                ;ax = 40*iy
536      push ax

```

```

537     mov ax,239
538     sub ax,dx             ;ix
539     mov bl,8
540     div bl                ;al = ix/8, ah = remainder
541     mov dl,ah
542     mov dh,0
543     mov ah,0
544     pop bx
545     push dx
546     add ax,bx             ;ax = 40*y+mod8(ix)
547     mov bl,al
548     push ax              ;address
549     call disdat
550     pop ax
551     push ax
552     mov bl,ah
553     call disdat
554     mov bl,00100100b     ;address pointer set
555     call discmd
556     ;
557     mov bx,offset image
558     pop ax               ;address
559     add bx,ax
560     mov al,cs:[bx]      ;read image data
561     mov dl,al
562     ;
563     pop ax              ;remainder: 000 -> 111
564     ;
565     mov ah,10000000b
566     cmp al,0
567     jz setpd
568     ;
569     shr ah,1
570     cmp al,1
571     jz setpd
572     ;
573     shr ah,1
574     cmp al,2
575     jz setpd
576     ;
577     shr ah,1
578     cmp al,3
579     jz setpd
580     ;
581     shr ah,1
582     cmp al,4
583     jz setpd
584     ;
585     shr ah,1
586     cmp al,5
587     jz setpd
588     ;
589     shr ah,1
590     cmp al,6

```

```

591      jz  setpd
592      ;
593      shr ah,1
594      ;
595      setpd:  mov al,dl          ;image data
596      ret
597      stclrp  endp
598      ;
599      ;output data in bl to display
600      ;
601      disdat  proc
602      call chks01
603      mov ax,PIAPTA          ;load data
604      mov dx,ax
605      mov al,bl
606      out dx,al
607      mov ax,PIAPTB
608      mov dx,ax
609      mov ah,cs:b765dat
610      mov al,00001000b      ;B3 D
611      or al,ah
612      out dx,al
613      mov al,00001100b      ;B2 CE
614      or al,ah
615      out dx,al
616      mov al,00001101b      ;B0 W
617      or al,ah
618      out dx,al
619      mov al,00001100b      ;B0 W release
620      or al,ah
621      out dx,al
622      mov al,00001000b      ;B2 CE release, B3 D
623      or al,ah
624      out dx,al
625      ret
626      disdat  endp
627      ;
628      ;output command in bl to PTA to display
629      ;
630      discmd  proc
631      call chks01
632      mov ax,PIAPTA
633      mov dx,ax
634      mov al,bl
635      out dx,al
636      mov ax,PIAPTB
637      mov dx,ax
638      mov ah,cs:b765dat
639      mov al,00000000b      ;B3 /C
640      or al,ah
641      out dx,al
642      mov al,00000100b      ;B2 CE
643      or al,ah
644      out dx,al

```

```

645      mov al,00000101b      ;B0 W
646      or al,ah
647      out dx,al
648      mov al,00000100b      ;B0 W release
649      or al,ah
650      out dx,al
651      mov al,00000000b      ;B2 CE release
652      or al,ah
653      out dx,al
654      nop
655      nop
656      nop
657      mov al,00001000b      ;B3 D
658      or al,ah
659      out dx,al
660      ret
661      discmd   endp
662      ;
663      ;check status STA0/STA1
664      ;
665      chks01   proc
666      mov ax,PIACTR
667      mov dx,ax
668      mov al,10010000b      ;mode 0, PTA in, PTB out
669      out dx,al
670      chks01!: mov ax,PIAPT
671      mov dx,ax
672      mov ah,cs:b765dat
673      mov al,00000000b      ;B3 /C
674      or al,ah
675      out dx,al
676      mov al,00000100b      ;B2 CE
677      or al,ah
678      out dx,al
679      mov al,00000110b      ;B1 R
680      or al,ah
681      out dx,al
682      mov ax,PIAPTA
683      mov dx,ax
684      in al,dx              ;read A0 A1
685      mov bh,al
686      mov ax,PIAPT
687      mov dx,ax
688      mov ah,cs:b765dat
689      mov al,00000100b      ;B1 R release
690      or al,ah
691      out dx,al
692      mov al,00000000b      ;B2 CE release
693      or al,ah
694      out dx,al
695      and bh,3              ;A0 STA0 A1 STA1
696      jz  chks01l
697      ;
698      mov al,00001000b      ;B3 D

```

```

699      or al,ah
700      out dx,al
701      mov ax,PIACTR
702      mov dx,ax
703      mov al,10000000b      ;mode 0, PTA out, PTB out
704      out dx,al
705      chks01x: ret
706      chks01 endp
707      ;
708      ;check status STA3
709      ;
710      chks3 proc
711      mov ax,PIACTR
712      mov dx,ax
713      mov al,10010000b      ;mode 0, PTA in, PTB out
714      out dx,al
715      chks3l: mov ax,PIAPT
716      mov dx,ax
717      mov ah,cs:b765dat
718      mov al,00000000b      ;B3 /C
719      or al,ah
720      out dx,al
721      mov al,00000100b      ;B2 CE
722      or al,ah
723      out dx,al
724      mov al,00000110b      ;B1 R
725      or al,ah
726      out dx,al
727      mov ax,PIAPTA
728      mov dx,ax
729      in al,dx      ;read STA3
730      mov bh,al
731      mov ax,PIAPT
732      mov dx,ax
733      mov ah,cs:b765dat      ;*****here mov ah,al
734      mov al,00000100b      ;B1 R release
735      or al,ah
736      out dx,al
737      mov al,00000000b      ;B2 CE release
738      or al,ah
739      out dx,al
740      and bh,8      ;A3 STA3
741      jz chks3l
742      ;
743      mov al,00001000b      ;B3 /D
744      or al,ah
745      out dx,al
746      mov ax,PIACTR
747      mov dx,ax
748      mov al,10000000b      ;mode 0, PTA out, PTB out
749      out dx,al
750      chks3x: ret
751      chks3 endp
752      ;

```

```

753      ;                      draw or erase common code
754      drwera  proc
755      mov  ax,cs:xpos      ;x1 is last xpos
756      mov  cs:x1,ax
757      mov  bx,cs:ypos      ;y1 is last ypos
758      mov  cs:y1,bx
759      cmp  ax,cs:x2      ;x1=x2?
760      jnz  drwerb
761      cmp  bx,cs:y2      ;y1=y2?
762      jnz  drwerb
763      jmp  drwext      ;exit if both equal
764      drwerb:  mov  ax,cs:x2      ;xd=x2-x1
765      sub  ax,cs:x1
766      mov  cs:xd,ax
767      mov  cs:xinc,1      ;xinc=+1
768      jns  pxddrw      ;xd>=0?
769      neg  cs:xd      ;xd=-xd
770      neg  cs:xinc      ;xinc=-1
771      pxddrw:  mov  bx,cs:y2      ;yd=y2-y1
772      sub  bx,cs:y1
773      mov  cs:yd,bx
774      mov  cs:yinc,1      ;yinc=+1
775      jns  pyddrw      ;yd>=0
776      neg  cs:yd      ;yd=-yd
777      neg  cs:yinc      ;yinc=-1
778      pyddrw:  mov  ax,cs:yd      ;yd2=2*yd
779      shl  ax,1
780      mov  cs:yd2,ax
781      mov  ax,cs:xd      ;xd2=2*xd
782      shl  ax,1
783      mov  cs:xd2,ax
784      mov  ax,cs:xd      ;yd>xd?
785      cmp  cs:yd,ax
786      jg   pydxd
787      ;
788      ;                      xd>=yd
789      mov  ax,cs:yd2      ;err=2*yd-xd
790      sub  ax,cs:xd
791      mov  cs:err,ax
792      drwpla:  mov  dx,cs:x1      ;plot x1,y1
793      mov  ax,cs:y1
794      cmp  cs:drweraf,0      ;check draw or erase flag
795      jne  drwa
796      call  dispera
797      jmp  drwar
798      drwa:    call  dispplt
799      drwar:  mov  ax,cs:xinc      ;x1=x1+xinc
800      add  cs:x1,ax
801      mov  ax,cs:x2      ;x1=x2?
802      cmp  cs:x1,ax
803      je   drwexl
804      mov  ax,cs:yd2      ;err=err+2*yd
805      add  cs:err,ax
806      cmp  cs:err,0      ;err<=0?

```

```

807     jle drwpla
808     mov ax,cs:yinc      ;y1=y1+yinc
809     add cs:y1,ax
810     mov ax,cs:xd2       ;err=err-2*xd
811     sub cs:err,ax
812     jmp drwpla
813     ;
814     drwexl: jmp drwext
815     ;
816     ;                      yd>xd
817     pydx:  mov ax,cs:xd2   ;err=2*xd-yd
818     sub ax,cs:yd
819     mov cs:err,ax
820     drwplb: mov dx,cs:x1    ;plot x1,y1
821     mov ax,cs:y1
822     cmp cs:drweraf,0      ;check draw or erase flag
823     jne drwb
824     call dispera
825     jmp drwbr
826     drwb:  call dispplt
827     drwbr: mov ax,cs:yinc   ;y1=y1+yinc
828     add cs:y1,ax
829     mov ax,cs:y2          ;y1=y2?
830     cmp cs:y1,ax
831     je drwext
832     mov ax,cs:xd2         ;err=err+2*xd
833     add cs:err,ax
834     cmp cs:err,0          ;err<=0?
835     jle drwplb
836     mov ax,cs:xinc        ;x1=x1+xinc
837     add cs:x1,ax
838     mov ax,cs:yd2         ;err=err-2*yd
839     sub cs:err,ax
840     jmp drwplb
841     ;
842     drwext: mov dx,cs:x2    ;save x2 as xpos
843     mov cs:xpos,dx
844     mov ax,cs:y2          ;save y2 as ypos
845     mov cs:ypos,ax
846     cmp cs:drweraf,0      ;check draw or erase flag
847     jne drwexa
848     call dispera
849     jmp drwexr
850     drwexa: call dispplt
851     drwexr: ret
852     drwera endp
853     ;
854     dispplt proc
855     call stclrp            ;ix->dx, iy->ax
856     or al,ah
857     mov cs:[bx],al        ;write image data
858     ;
859     mov bl,al             ;display
860     call disdat

```

```

861      mov bl,11000000b      ;write data
862      call discmd
863      ret
864      dispplt endp
865      ;
866      dispera proc
867      call stclrp
868      xor al,ah
869      mov cs:[bx],al      ;write image data
870      ;
871      mov bl,al      ;display
872      call disdat
873      mov bl,11000000b      ;write data
874      call discmd
875      ret
876      dispera endp
877      ;
878      chdata dw ?
879      cdata dw ?
880      pdata dw ?
881      rdata dw ?
882      xpos dw ?
883      ypos dw ?
884      xpos dw ?
885      ypos dw ?
886      stccnt dw ?
887      xinc dw ?
888      xd dw ?
889      xd2 dw ?
890      x1 dw ?
891      x2 dw ?
892      yinc dw ?
893      yd dw ?
894      yd2 dw ?
895      y1 dw ?
896      y2 dw ?
897      err dw ?
898      drweraf dw ?
899      ;
900      image db 5120 dup (0)
901      ;
902      chrdata db 00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,7EH,00H;CUR
903      db 00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H,00H;NULL
904      db 255 dup(0)
905      db 179 dup(0)
906      db 00H,10H,10H,10H,10H,10H,10H,00H,10H,00H,00H,00H,00H,00H ;!
907      db 00H,28H,28H,28H,28H,00H,00H,00H,00H,00H,00H,00H,00H,00H ;"
908      db 00H,00H,28H,28H,7CH,28H,28H,7CH,28H,28H,00H,00H,00H,00H ;#
909      db 00H,10H,38H,54H,50H,30H,18H,14H,54H,38H,10H,00H,00H,00H ;$
910      db 00H,64H,64H,08H,08H,10H,20H,20H,4CH,4CH,00H,00H,00H,00H ;%
911      db 00H,30H,48H,48H,50H,20H,54H,4CH,48H,34H,00H,00H,00H,00H ;&
912      db 00H,10H,10H,10H,20H,00H,00H,00H,00H,00H,00H,00H,00H,00H ;'
913      db 00H,08H,10H,10H,10H,10H,10H,10H,10H,08H,00H,00H,00H,00H ;(
914      db 00H,20H,10H,10H,10H,10H,10H,10H,10H,20H,00H,00H,00H,00H ;)

```

```

915 db 00H,00H,10H,54H,38H,10H,38H,54H,10H,00H,00H,00H,00H ;*
916 db 00H,00H,00H,10H,10H,7CH,10H,10H,00H,00H,00H,00H,00H ;+
917 db 00H,00H,00H,00H,00H,00H,00H,00H,10H,10H,10H,20H,00H,00H ;,
918 db 00H,00H,00H,00H,00H,00H,7CH,00H,00H,00H,00H,00H,00H ;-
919 db 00H,00H,00H,00H,00H,00H,00H,00H,10H,10H,00H,00H,00H,00H ;.
920 db 00H,04H,04H,08H,08H,10H,20H,20H,40H,40H,00H,00H,00H,00H ;/
921 db 00H,38H,44H,44H,4CH,54H,64H,44H,44H,38H,00H,00H,00H,00H ;0
922 db 00H,10H,30H,10H,10H,10H,10H,10H,10H,38H,00H,00H,00H,00H ;1
923 db 00H,38H,44H,04H,08H,10H,20H,40H,40H,7CH,00H,00H,00H,00H ;2
924 db 00H,38H,44H,04H,04H,18H,04H,04H,44H,38H,00H,00H,00H,00H ;3
925 db 00H,08H,08H,18H,28H,48H,48H,7CH,08H,08H,00H,00H,00H,00H ;4
926 db 00H,7CH,40H,40H,78H,44H,04H,04H,44H,38H,00H,00H,00H,00H ;5
927 db 00H,18H,24H,40H,40H,78H,44H,44H,44H,38H,00H,00H,00H,00H ;6
928 db 00H,7CH,44H,04H,08H,08H,10H,10H,10H,10H,00H,00H,00H,00H ;7
929 db 00H,38H,44H,44H,44H,38H,44H,44H,44H,38H,00H,00H,00H,00H ;8
930 db 00H,38H,44H,44H,44H,3CH,04H,04H,48H,30H,00H,00H,00H,00H ;9
931 db 00H,00H,00H,10H,10H,00H,00H,00H,10H,10H,00H,00H,00H,00H ;:
932 db 00H,00H,00H,10H,10H,00H,00H,00H,10H,10H,10H,20H,00H,00H ;;
933 db 00H,00H,00H,08H,10H,20H,10H,08H,00H,00H,00H,00H,00H,00H ;<
934 db 00H,00H,00H,00H,7CH,00H,7CH,00H,00H,00H,00H,00H,00H,00H ;=
935 db 00H,00H,00H,20H,10H,08H,10H,20H,00H,00H,00H,00H,00H,00H ;>
936 db 00H,38H,44H,04H,08H,10H,10H,00H,10H,10H,00H,00H,00H,00H ;?
937 db 00H,38H,44H,4CH,54H,54H,54H,4CH,40H,3CH,00H,00H,00H,00H ;@
938 db 00H,10H,28H,44H,44H,7CH,44H,44H,44H,44H,00H,00H,00H,00H ;A
939 db 00H,78H,24H,24H,24H,38H,24H,24H,24H,78H,00H,00H,00H,00H ;B
940 db 00H,38H,44H,40H,40H,40H,40H,44H,38H,00H,00H,00H,00H ;C
941 db 00H,78H,24H,24H,24H,24H,24H,24H,78H,00H,00H,00H,00H ;D
942 db 00H,7CH,24H,20H,28H,38H,28H,20H,24H,7CH,00H,00H,00H,00H ;E
943 db 00H,7CH,24H,20H,28H,38H,28H,20H,20H,70H,00H,00H,00H,00H ;F
944 db 00H,38H,44H,40H,40H,4CH,44H,44H,4CH,34H,00H,00H,00H,00H ;G
945 db 00H,44H,44H,44H,44H,7CH,44H,44H,44H,44H,00H,00H,00H,00H ;H
946 db 00H,38H,10H,10H,10H,10H,10H,10H,10H,38H,00H,00H,00H,00H ;I
947 db 00H,3CH,08H,08H,08H,08H,08H,48H,48H,30H,00H,00H,00H,00H ;J
948 db 00H,64H,24H,24H,28H,30H,28H,24H,24H,64H,00H,00H,00H,00H ;K
949 db 00H,70H,20H,20H,20H,20H,20H,20H,7CH,00H,00H,00H,00H ;L
950 db 00H,44H,44H,6CH,54H,44H,44H,44H,44H,00H,00H,00H,00H ;M
951 db 00H,44H,44H,44H,64H,54H,4CH,44H,44H,44H,00H,00H,00H,00H ;N
952 db 00H,38H,44H,44H,44H,44H,44H,44H,38H,00H,00H,00H,00H ;O
953 db 00H,78H,24H,24H,24H,38H,20H,20H,20H,70H,00H,00H,00H,00H ;P
954 db 00H,38H,44H,44H,44H,44H,44H,54H,4CH,38H,04H,00H,00H,00H ;Q
955 db 00H,78H,24H,24H,24H,38H,28H,24H,24H,64H,00H,00H,00H,00H ;R
956 db 00H,38H,44H,40H,20H,10H,08H,04H,44H,38H,00H,00H,00H,00H ;S
957 db 00H,7CH,54H,10H,10H,10H,10H,10H,10H,10H,00H,00H,00H,00H ;T
958 db 00H,44H,44H,44H,44H,44H,44H,44H,38H,00H,00H,00H,00H ;U
959 db 00H,44H,44H,44H,44H,44H,44H,28H,10H,00H,00H,00H,00H ;V
960 db 00H,44H,44H,44H,44H,44H,54H,54H,54H,28H,00H,00H,00H,00H ;W
961 db 00H,44H,44H,44H,28H,10H,28H,44H,44H,44H,00H,00H,00H,00H ;X
962 db 00H,44H,44H,44H,28H,10H,10H,10H,10H,38H,00H,00H,00H,00H ;Y
963 db 00H,7CH,44H,04H,08H,10H,20H,40H,44H,7CH,00H,00H,00H,00H ;Z
964 db 00H,38H,20H,20H,20H,20H,20H,20H,38H,00H,00H,00H,00H ;[
965 db 00H,40H,40H,20H,20H,10H,08H,08H,04H,04H,00H,00H,00H,00H ;\
966 db 00H,38H,08H,08H,08H,08H,08H,08H,38H,00H,00H,00H,00H ;]
967 db 00H,00H,00H,00H,10H,28H,44H,00H,00H,00H,00H,00H,00H,00H ;^
968 db 00H,00H,00H,00H,00H,00H,00H,00H,7CH,00H,00H,00H,00H ;_

```

```

969 db 00H,10H,10H,10H,08H,00H,00H,00H,00H,00H,00H,00H,00H ;`
970 db 00H,00H,00H,00H,38H,48H,48H,48H,48H,34H,00H,00H,00H ;a
971 db 00H,60H,20H,20H,38H,24H,24H,24H,24H,78H,00H,00H,00H ;b
972 db 00H,00H,00H,00H,38H,44H,40H,40H,44H,38H,00H,00H,00H ;c
973 db 00H,18H,08H,08H,38H,48H,48H,48H,48H,34H,00H,00H,00H ;d
974 db 00H,00H,00H,00H,38H,44H,7CH,40H,44H,38H,00H,00H,00H ;e
975 db 00H,08H,14H,10H,10H,38H,10H,10H,10H,10H,00H,00H,00H ;f
976 db 00H,00H,00H,00H,34H,48H,48H,38H,20H,38H,44H,44H,38H ;g
977 db 00H,60H,20H,20H,38H,24H,24H,24H,24H,64H,00H,00H,00H ;h
978 db 00H,30H,00H,00H,30H,10H,10H,10H,10H,38H,00H,00H,00H ;i
979 db 00H,18H,00H,00H,18H,08H,08H,08H,08H,08H,28H,10H,00H ;j
980 db 00H,60H,20H,20H,24H,28H,30H,28H,24H,64H,00H,00H,00H ;k
981 db 00H,30H,10H,10H,10H,10H,10H,10H,10H,38H,00H,00H,00H ;l
982 db 00H,00H,00H,00H,68H,54H,54H,54H,44H,44H,00H,00H,00H ;m
983 db 00H,00H,00H,00H,58H,64H,44H,44H,44H,44H,00H,00H,00H ;n
984 db 00H,00H,00H,00H,38H,44H,44H,44H,44H,38H,00H,00H,00H ;o
985 db 00H,00H,00H,00H,58H,24H,24H,24H,34H,28H,20H,20H,60H ;p
986 db 00H,00H,00H,00H,34H,48H,48H,48H,58H,28H,08H,08H,0CH ;q
987 db 00H,00H,00H,00H,58H,24H,20H,20H,20H,70H,00H,00H,00H ;r
988 db 00H,00H,00H,00H,38H,44H,30H,08H,44H,38H,00H,00H,00H ;s
989 db 00H,10H,10H,10H,38H,10H,10H,10H,14H,08H,00H,00H,00H ;t
990 db 00H,00H,00H,00H,48H,48H,48H,48H,48H,34H,00H,00H,00H ;u
991 db 00H,00H,00H,00H,44H,44H,44H,44H,28H,10H,00H,00H,00H ;v
992 db 00H,00H,00H,00H,44H,44H,54H,54H,54H,28H,00H,00H,00H ;w
993 db 00H,00H,00H,00H,44H,28H,10H,10H,28H,44H,00H,00H,00H ;x
994 db 00H,00H,00H,00H,44H,44H,44H,24H,1CH,04H,04H,48H,30H ;y
995 db 00H,00H,00H,00H,7CH,04H,08H,10H,20H,7CH,00H,00H,00H ;z
996 db 00H,08H,10H,10H,10H,20H,10H,10H,10H,08H,00H,00H,00H ;{
997 db 00H,10H,10H,10H,10H,00H,10H,10H,10H,10H,00H,00H,00H ;|
998 db 00H,20H,10H,10H,10H,08H,10H,10H,10H,20H,00H,00H,00H ;}
999 db 00H,00H,00H,00H,00H,34H,48H,00H,00H,00H,00H,00H,00H ;~
1000 ;
1001 b765dat db ?
1002 ;
1003 end
1004 ;
1005 ;end of helmsub.asm

```